PHILIPS

PHILIPS

PHILIPS MSX HOME COMPUTER SYSTEM



MANUAL DEL USUARIO

MSX

PHILIPS

PHILIPS MSX
HOME COMPUTER SYSTEM

N.º USUARIO
9.834

Estimado cliente:

Gracias por la confianza depositada en nosotros al haber adquirido un PHILIPS MSX 'HOME-COMPUTER' SYSTEM; desde este momento toda una organización le prestará el apoyo necesario para que pueda obtener el máximo beneficio de su compra.

Antes de llegar a su poder, este ordenador ha superado con éxito los más estrictos controles de calidad y su buen funcionamiento estará respaldado por la amplia red de servicios Philips.

Por favor, envíenos la respuesta adjunta con sus datos y número de usuario PHILIPS MSX. Esto nos permitirá poder procesar los datos, con el objeto de prestarle un buen servicio en todas las consultas que desee formularnos. Nosotros, en contrapartida, le mantendremos informado de todo aquello que pueda interesarle por estar relacionado con su PHILIPS MSX —nuevos periféricos, programas de aplicación, documentación informática, ofertas club de usuario, etc.—.

Nuestra intención no es otra que la de mantener una comunicación viva con el usuario, entendiéndola como un aspecto más de nuestros servicios.

R. AVILA

Jefe de Producto Home Computer
PHILIPS IBERICA, S.A.E.

RED DE SERVICIOS PHILIPS
TALLERES PROPIOS DE SERVICIO

MADRID
Albasanz, 75.
Teléfs.: 204 59 40/204 70 25

Vallehermoso, 14.
Teléf.: 448 05 63

BARCELONA
Bruch, 147-149
Teléf.: (93) 207 02 62.

LAS PALMAS
Alfred Nobel, 8 (Los Tarahales).
Teléf.: (928) 25 52 40

SEVILLA
Feria, 164.
Teléf.: (954) 37 26 16

BILBAO
José Olabarría, 6
Teléf.: (94) 432 14 50

VALENCIA
Dr. Juan Regla, 6
(frente al Estadio Luis Casanova).
Teléf.: (96) 362 53 61

TALLERES AUTORIZADOS INDEPENDIENTES

SUCURSAL MADRID

ELECTRONICA SANCHIS
Fe Moreno Hernán
Zarza, 4
13003 CIUDAD REAL

TV. BOUZA
Eduardo Bouza Pérez
La Venta, 59
COLLADO VILLALBA (Madrid)

Eduardo Barambio Bayo
Reyes Católicos, 36
16003 CUENCA

MORASA, S.A.
Pedro Pascual, 15
19001 GUADALAJARA

Francisco Guijo Oliva
Ramón y Cajal, 1
40002 SEGOVIA

Francisco Díaz Machuca
Mesones, 44
TALAVERA DE LA REINA
(Toledo)

FECAR
Manuel Martínez Gómez
Plaza Santa Clara, 4
45002 TOLEDO

ZARAGOZA SERVICIO, S.A.
Madre Vedruna, 8
50008 ZARAGOZA

SUCURSAL BILBAO

CASTELE
Luis Ilarduya Los Arcos
Nuestra Señora de Belén, 6
09001 BURGOS

ELECTRONICA CADARSO
Miguel Cadarso Ruiz
Primo de Rivera, 3 bajo
26001 LOGROÑO

ANTHER ELECTRONICA
José Moreno Goñi
Avda. Isabel II, 23
20011 SAN SEBASTIAN

CENTRO ELECTRONICO MONTAÑES
Luis Martínez Castanedo
Isaac Peral, 28
39008 SANTANDER

Javier Arbé Echarri
Monasterio de Irache, 28
31011 PAMPLONA

AMERICA ELECTRO RADIO
Enrique Casi Vicente
Beato Tomás de Zumárraga, 19
01008 VITORIA

SUCURSAL VALENCIA

SELVITEL
Rosario, 131
02003 ALBACETE

TALLERES FLORIDA
Juan Jimeno Peris
Grado, 14
03007 ALICANTE

ELECTROCOLOR
Avelino Marín López
Juan Fernández, 39
CARTAGENA (Murcia)

CRISON, S.L.
Jorge Juan, 93
12006 CASTELLON

José Herrerías Castilla
Avda. de la Libertad, 26
ELCHE (Alicante)

Vicente Della Naquer
Poeta Llorente, 10
GANDIA (Valencia)

Eduardo Francés Sanchís
Pere IV Aragón, 10
JATIVA (Valencia)

MURCIA SERVICIO
Mariano Vergara, 9
30003 MURCIA

Joaquín Haro Pascual
General Ricardo Ortega, 26 A
07006 PALMA DE MALLORCA

SUCURSAL CANARIAS

José A. Rodríguez Reyes
Buena Vista, 45
BREÑA ALTA - La Palma
(I. Canarias)

TENERIFE SERVICIO, S.A.
Alcalde Mandillo Tejera, s/n
38007 SANTA CRUZ TENERIFE

SUCURSAL NOROESTE

ELECTRONICA ELPE
Elio Fernández Fernández
La Coruña, 11
BARCO DE VALDEORRAS
(Orense)

HERMANOS RODRIGUEZ
José J. Rodríguez Gómez
Os Fidalgo Irma Denis, 17
CARBALLINO (Orense)

Delfín Espino Faraldo
P. Camino de los Cuervos, 1
EL FERROL (La Coruña)

COPHIL SDAD. COOP. LTDA.
Pintor Seijo Rubio, 17
15006 LA CORUÑA

Lorenzo Fernández Fernández
Avda. Alvaro López Núñez, 10
24002 LEON

ELECTRONICA LUCENSE
Víctor Rey González
Profesor Otero Pedrayo, 87
27003 LUGO

ALFRAN RADIO
Francisco Fernández López
Calvo Sotelo, 66 bajo
MORFORTE DE LEMOS (Lugo)

ELECTRONICA J. FERNANDEZ
José Fernández González
Avda. Buenos Aires, 66
32004 ORENSE

Ignacio Faya Faya
Monte del Sueve, 33
33012 OVIEDO

Daniel Hernández Saiz
Juan de Argüelles, 21
37004 SALAMANCA

ELECTRONICA SANCHEZ
Antonio Sánchez Lápido
Transperegrina bloque 4
SANTIAGO DE COMPOSTELA
(La Coruña)

VALLADOLID SERVICIO, S.A.
Carretera de la Esperanza, 83
47008 VALLADOLID

SERVITEC
Domingo Martínez Rajo
Queipo de Llano, 31
VIGO (Pontevedra)

VIGOSERVIC, S.L.
El Pino, 56
VIGO (Pontevedra)

Modesto Martínez Costa
Río Dos Foles, 16 bajo
VIVERO (Lugo)

SUCURSAL SEVILLA

GIBRALCAMPO
Ramón Tell Lozano
Bahía Algeciras, bloque 11
ALGECIRAS (Cádiz)

ELECTRO BORCHERS
Gunther Borchers
Zaragoza, 14
04001 ALMERIA

Manuel Torres Macías
Encarnación, 8
ANTEQUERA (Málaga)

Manuel Bonifacio Jiménez
Felipe Checa, 16
06001 BADAJOZ

MANUEL ROS GARCIA
Monjas, 10
BAZA (Granada)

TELETECNO
José Martín Baltasar
Calderón de la Barca, 6
11003 CADIZ

SERVICOLOR
Felipe Pizarro Valle
Juan XXIII, 4 bajo
10001 CACERES

Eduardo Francés Cerrillo
Conde de Vallellano, 21
14004 CORDOBA

Francisco Martín Sánchez
Chorro, 41
ESTEPONA (Málaga)

ELECTRONICA 2000, S.L.
Miguel Bueno, 1
FUENGIROLA (Málaga)

GRANADA SERVICIO, S.A.
Méndez Núñez, 4
18003 GRANADA

TELESONID
Manuel Silva López
Francisco Vázquez Limón, 4
21002 HUELVA

PHILSERVIC
Tomás Cano Rubio
Avda. Barcelona, 29
23006 JAEN

ROFER
Francisco Romero Rubio
Doctrina, 28
JEREZ DE LA FRONTERA
(Cádiz)

Tomás Rubio de Dios
Hernán Cortés, 13
LINARES (Jaén)

MALAGA SERVICIO, S.A.
Ayala, 9
29002 MALAGA

TELEMAR
Antonio Pacheco Pérez
Boquerón, local 7
MARBELLA (Málaga)

ELECTRONICA MABEDA
Manuel Morales Gutiérrez
Badajoz, bloque 4
MERIDA (Badajoz)

Faustino Hernández Clemente
Concepción, 10
MERIDA (Badajoz)

Francisco Galeote Romero
Cardenal Belluga, 10
MOTRIL (Granada)

Julián Serrano Fuentes
Cartas, 4
PLASENCIA (Cáceres)

Bartolomé Pacheco Pérez
Castro Sandaza, 1
RONDA (Málaga)

MANUAL DEL USUARIO
PHILIPS MSX
HOME COMPUTER SYSTEM

Este manual ha sido publicado por Philips Export B.V., Corporate Centre Video, Eindhoven, Holland.

Versión en castellano

Traduce, compone e imprime: CONORG, S.A.

Supervisión Técnica: PHILIPS IBERICA, S.A.E.

I.S.B.N.: 84-398-2583-8
Depósito Legal: M-36.600-1984

Contenido

Introducción

Ahora tienes un 'home computer' moderno y versátil que puedes usarlo para aprender a escribir tus propios programas informáticos, desde el más simple, avanzando hasta el más complejo. Puede usarse también para jugar con un amplio surtido de programas existentes en el mercado.

El ordenador satisface las últimas normas para 'home computer': el estándar MSX. Te ofrece una ventaja importante, dado que significa que puedes aprovecharte de cualquiera de los programas disponibles en la extensa programoteca MSX. También significa que tienes una amplia elección de equipos periféricos, tales como impresoras, unidades de disco, etc.; que pueden usarse en combinación con el 'home computer'.

El ordenador tiene incorporado un intérprete MSX-BASIC, compatible con el de otros ordenadores personales MSX. El MSX-BASIC reconoce casi todas las instrucciones de la versión Microsoft MBASIC-80, además de instruciones para generación de música, presentación en colores, "sprites", y el uso de "joysticks" para juegos.

En este manual, encontrarás las instrucciones para instalar tu 'home computer'.

El Capítulo 2 explica el significado y el uso de las diversas instrucciones MSX-BASIC. Aunque encontrarás una descripción detallada en el manual de referencia.

El capítulo 3 ("Hardware") te dice cómo ampliar las posibilidades de tu 'home computer'.

En el Capítulo 4 ("Apéndices") encontrarás importante información adicional.

Si lees cuidadosamente este manual, ensayando con todos estos ejemplos reflejados, pronto serás capaz de escribir tus propios programas en el ordenador. Te deseamos que disfrutes completamente con tu compacto, aunque increíblemente versátil, 'home computer'.

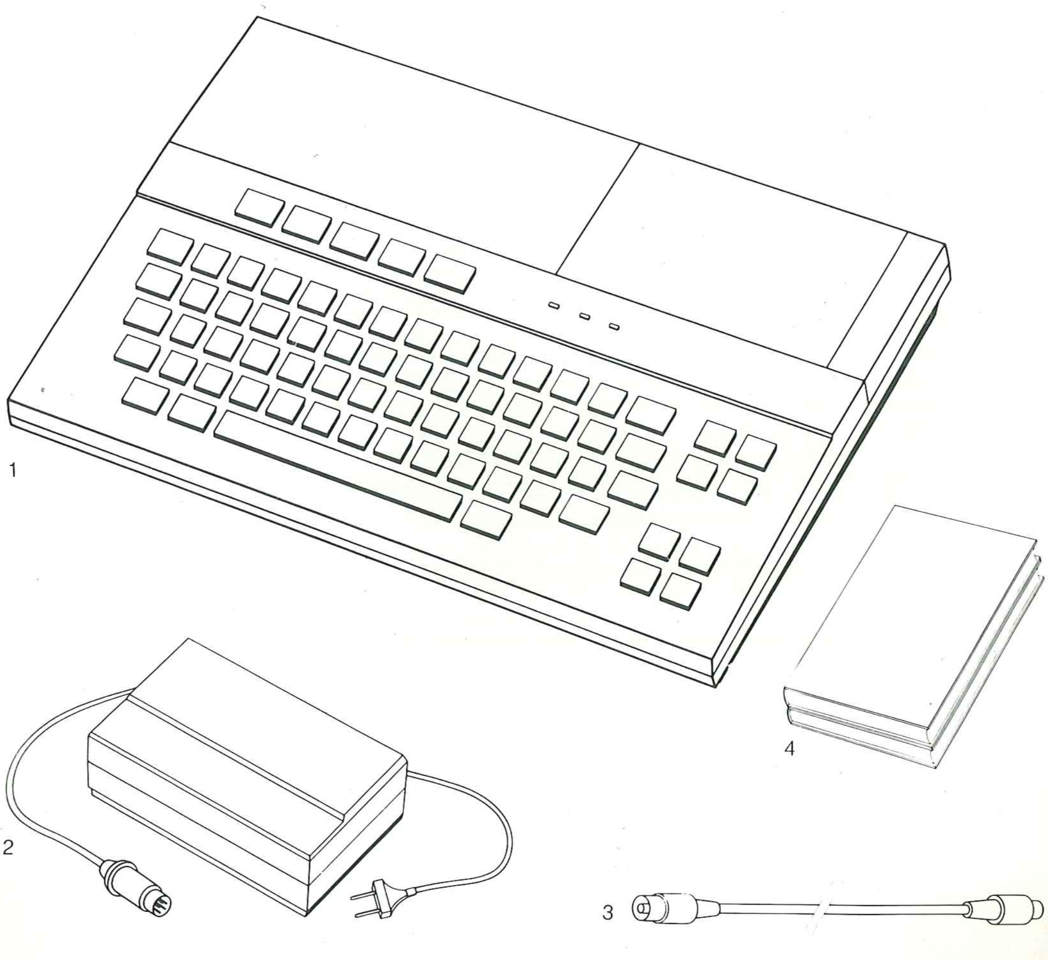
Instalación del 'home computer'

Capítulo 1

Desembalaje

Capítulo 1.1. Cuando abras el embalaje, encontrarás todo lo que necesitas para iniciar la operación con tu MSX 'home computer'. Para asegurarte que está completo, comprueba el contenido. El embalaje deberá contener los siguientes artículos:

1. La consola 'home computer' con su teclado alfanumérico.
2. Una fuente de alimentación para dotar a tu ordenador de la energía eléctrica apropiada.
3. Un cable de conexión a tu receptor de televisión.
4. Los manuales 'home computer'.



Para usar tu 'home computer', necesitas sólo un receptor de TV en blanco y negro o en color; o un monitor de color para sacar el máximo beneficio de las amplias posibilidades de color que ofrece el ordenador.

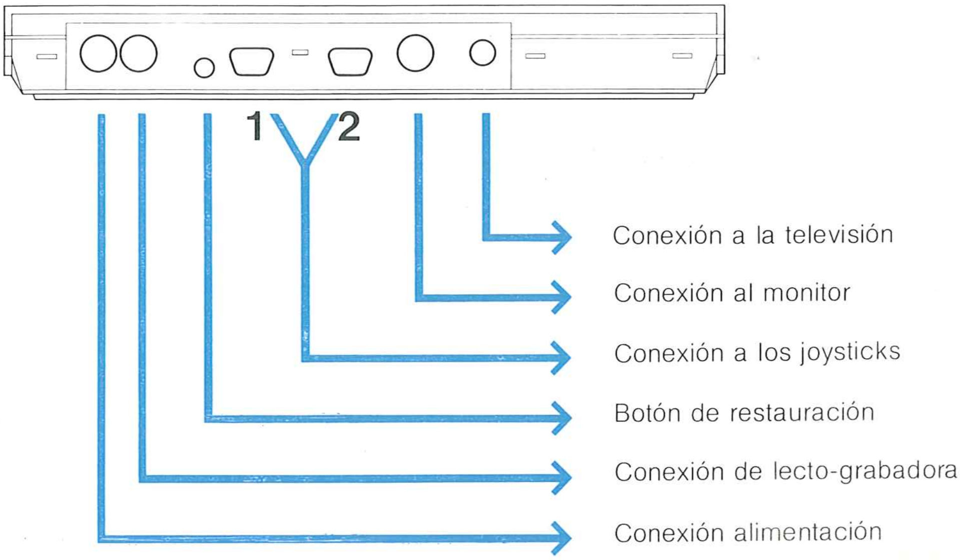
Sigue las instrucciones de instalación en el siguiente capítulo 1.2 y eso es todo lo que necesitas para comenzar.

Hay un amplio surtido de accesorios especiales conectables a tu MSX 'home computer', que amplían sus posibilidades.

1. Una **lecto-grabadora de datos,** que puede usarse como memoria externa para almacenamiento de programas y ficheros en cassette.
2. **Joysticks** de juegos, para disfrutar con los muchos e interesantes juegos de video por ordenador disponibles para el MSX 'home computer'.
3. Un **monitor de video** que te da el beneficio de imágenes más claras y nítidas... en cuanto a la TV.
4. Un **cartucho** de ampliación para aumentar la memoria de tu 'home computer'.
5. Una **impresora** sobre papel para producir copias impresas de tus programas y ficheros. La combinación de la impresora y un programa de tratamiento de textos, puedes aprovecharla para escribir cartas, informes, etc.
6. Diversos **cartuchos** con programas especiales.
7. **Unidades de diskettes** como memoria externa de gran capacidad y fácilmente accesible para conservar programas y ficheros.

Ajustando las conexiones

Capítulo 1.2 En la parte trasera de tu ordenador encontrarás los siguientes enchufes de conexión:



Ajuste de las conexiones

1. Conecta primero el cable especial al enchufe TV de tu ordenador, luego conecta el otro extremo a la entrada de antena en tu aparato de televisión.
2. Conecta la fuente de alimentación al enchufe de entrada de energía de ordenador, luego conéctalo a la red eléctrica.
3. Acciona ahora el interruptor del ordenador (marcado on/off) de la fuente de alimentación.
4. Enciende tu receptor TV. Escoge uno de los canales de preselección de tu aparato que no sea usado para la recepción regular de TV y resérvalo para tu ordenador. Elige el canal correcto (el canal correcto se indica en una etiqueta del panel posterior de tu ordenador, al lado del conector correspondiente).
5. Verás aparecer en pantalla:

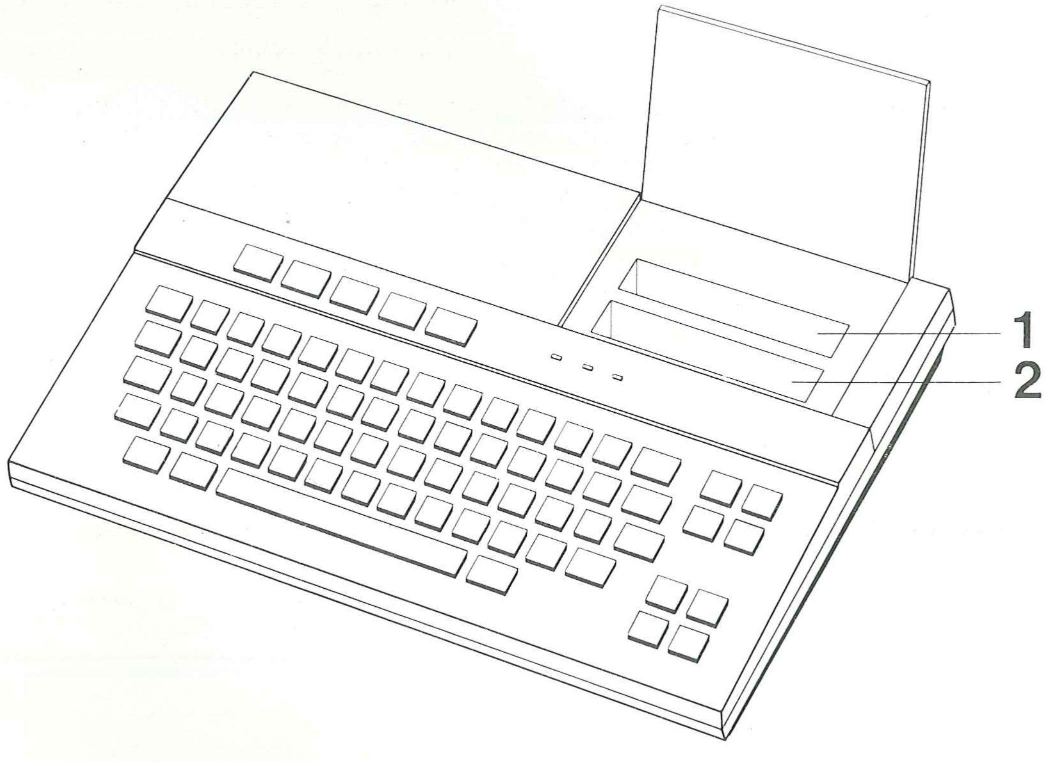
```
MSX system
etc.
```

Si usas un monitor o un receptor de color, la pantalla tendrá un fondo azul y las letras estarán en blanco.

6. Cuando no se ha insertado ningún cartucho, verás el siguiente texto en pantalla:

```
MSX BASIC
etc.
```

Tu 'home computer' ahora está **preparado** para ser usado.



Inserción del cartucho

APAGA SIEMPRE TU 'HOME COMPUTER' (OFF) ANTES DE INSERTAR UN CARTUCHO EN UNA DE LAS DOS RANURAS PARA CARTUCHOS.

Inserta el cartucho, con el lado de las etiquetas hacia ti, en una de las ranuras previstas. Luego, enciende de nuevo tu ordenador.

Conexión de los joysticks

Apaga el ordenador antes de conectarlos.

El 'home computer' tiene enchufes especiales para dos joysticks de juegos.

El mando n.º 1 es para el control con la mano derecha.

El mando n.º 2 es para el control con la mano izquierda.

Conexión de una lecto-grabadora de datos

Apaga el ordenador antes de conectar la registradora de datos.

ESO SIGNIFICA QUE DEBERAS CONECTAR LA LECTO-GRABADORA ANTES DE COMENZAR A ESCRIBIR UN PROGRAMA EN EL ORDENADOR. SI NO LO HACES, TODA LA INFORMACION CONSERVADA EN LA MEMORIA DEL ORDENADOR SE PERDERA EN EL MOMENTO EN QUE INTERRUMPAS LA ENERGIA ELECTRICA DE ALIMENTACION.

Recomendamos el uso de una lecto-grabadora de datos (registradora de cassette) con 'control remoto'. Cuando compres una registradora de cassette, asegúrate que tu distribuidor te suministra también el cable de conexión adecuado. Conecta este cable al enchufe correspondiente de tu ordenador. En la mayoría de los casos, encontrarás tres clavijas en el otro extremo. Estas deberán conectarse a tu lecto-grabadora de datos como sigue:

1. La clavija **negra** va en la patilla REM de la registradora de cassette.
2. La clavija **roja** va en la patilla MIC de la registradora de cassette.
3. La clavija **blanca** va en la patilla EAR de la registradora de cassette.

Para operar la lecto-grabadora, véase el Capítulo 2.5.

Conexión del monitor

Un enchufe especial está previsto en el ordenador para conectar un monitor. Un **monitor de video,** como ya sabrás, es similar a un receptor de TV pero no permite recibir los programas oficiales de la televisión. Sin embargo, producirá imágenes procedentes de una 'fuente directa', tales como el procesador de video de tu ordenador.

Cuando compres un monitor, asegúrate que el distribuidor también te suministra un cable de conexión adecuado.

Uso del botón RESET

Cuando pulsas el botón RESET, situado en el panel trasero del ordenador, la operación del ordenador se interrumpirá temporalmente, RESTAURANDO las condiciones iniciales. Eso significa que se borra completamente el contenido de la memoria y se prepara para un comienzo de operación, de la misma manera que cuando enciendes el ordenador por primera vez.

IMPORTANTE

Asegúrate que todas las clavijas están firmemente conectadas en los zócalos adecuados, antes de dar corriente a tu 'home computer'.

Para quitar la cubierta de los conectores para cartuchos, ábrela completamente, luego empújala ligeramente hacia atrás y quedará libre y suelta.

El teclado

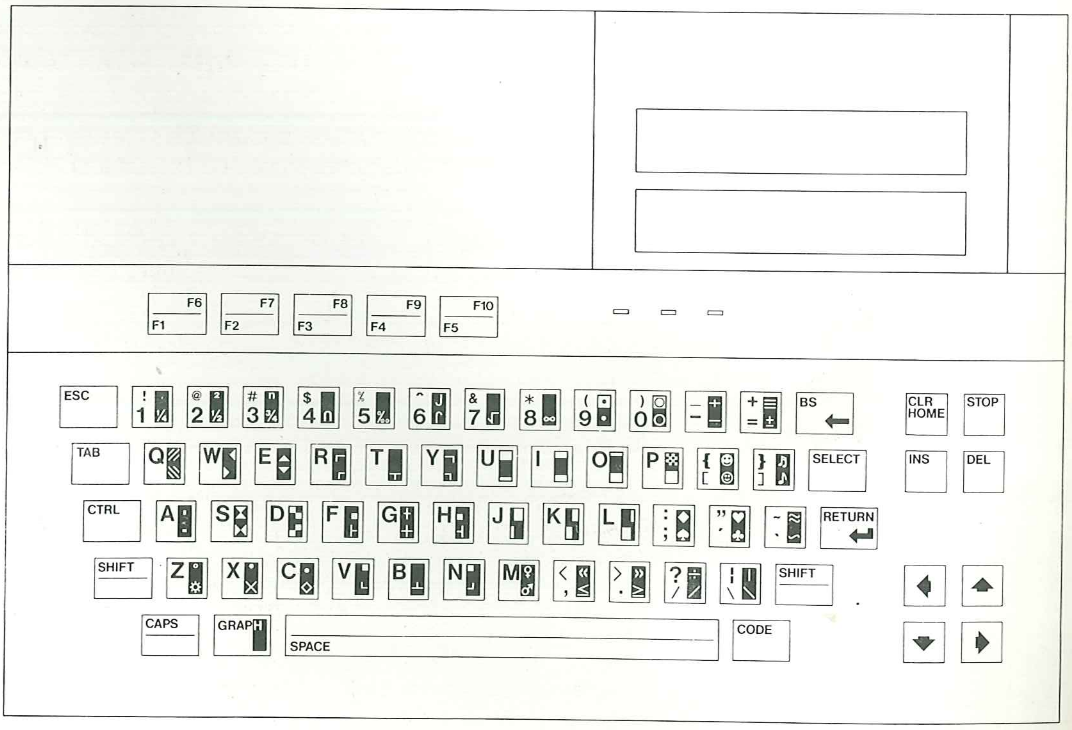
Capítulo 1.3

Avanza y ensaya todas las teclas; no le hará daño al ordenador, suponiendo desde luego que no empleas excesiva fuerza. Realmente necesitas muy poca fuerza para pulsar las teclas. Es buena idea ver con qué poco esfuerzo consigues que aparezcan los caracteres en la pantalla.

Si mantienes pulsada una tecla más tiempo del necesario, tendrá el mismo efecto que si repitieras la pulsación de la tecla continua y sucesivamente, hasta que dejes de pulsarla.

El teclado se compone de cuatro grupos de teclas. Comenzando por la parte superior verás:

1. Cinco **teclas funcionales** en la fila superior.
2. En el extremo superior derecho, verás cuatro **teclas de comando** "HOME", "STOP", "INS" Y "DEL"
3. Inmediatamente debajo de las anteriores, las cuatro **teclas de cursor**, para desplazarlo en cualquier dirección.
4. El resto de las teclas comprenden todas las habituales en las máquinas de escribir, junto con algunas teclas especiales tales como "ESC", "TAB", "CTRL", etc.



Teclas funcionales Las funciones reales de estas teclas son programables por el usuario (véase capítulo 2.1). Las teclas, sin embargo, están programadas previamente y las funciones ejercidas se indican en la parte inferior de la pantalla en el MSX-BASIC, a continuación de los indicativos de las teclas F1 a F5.

Si pulsas la tecla de turno (SHIFT), verás las funciones preprogramadas F6 a F10 en las mismas teclas. Pulsando por ejemplo simultáneamente la tecla SHIFT y la F1, se ejecuta la función programada para F6.

Teclas de comando Las teclas de comando ejercen en el ordenador las siguientes funciones:

HOME	El cursor se desplaza a la primera columna de la primera línea, que es la posición BASE.
HOME+SHIFT	Pulsando simultáneamente estas dos teclas se pasa el cursor a la posición base y se LIMPIA la pantalla.
STOP	Esta tecla se usa para hacer que PARE la ejecución de un programa. El programa reanudará la ejecución cuando se pulse de nuevo la misma tecla. De hecho ejerce la función de 'pausa'.
INS	Al pulsarla se pasa al modo de INSERCION, que permite insertar texto en el que aparece en pantalla. Se sale de ese modo de inserción, pulsando de nuevo la misma tecla (INS corresponde a "inserción").
DEL	Al pulsar esta tecla se quita del texto que aparece en pantalla, el carácter señalado en ese momento por el cursor. (DEL corresponde a "delectar", o sea, suprimir).

Teclas de control del cursor Las flechas indican la dirección en que se desplazará el cursor cuando se pulse una tecla de este grupo.

Teclas tipográficas Estas teclas funcionan exactamente igual a las de una máquina de escribir normal. Si pulsas la tecla marcada "A", se enviará hacia el ordenador el código correspondiente a la letra "a" que asimismo aparecerá en pantalla. Si pulsas esa misma tecla, manteniendo simultáneamente pulsada la tecla SHIFT, el símbolo enviado será la "A" mayúscula. Las otras teclas para cifras y signos de puntuación funcionan de la misma manera.

Rodeando las teclas tipográficas estándar, observarás también unas cuantas teclas especiales:

ESC	Esta tecla sólo ejerce su función de ESCAPE del código convenido, cuando hay conectada una impresora al 'home computer' (ESC corresponde a "escape").
TAB	Al pulsar esta tecla, el cursor avanzará hasta el siguiente tope de TABULACION (TAB corresponde a "tabular").
CTRL	Esta tecla sólo ejerce una función cuando se usa en combinación con otra tecla, como se describe en el Apéndice D (CTRL corresponde a "control").
SHIFT	La función de la tecla de TURNO, es elegir el símbolo inferior o superior de los marcados en cada tecla tipográfica.
CAPS	Cuando se pulsa esta tecla, se iluminará el piloto CAPS y se enviarán hacia el ordenador los caracteres correspondientes a las letras mayúsculas únicamente. Si vuelves a pulsar de nuevo la misma tecla, las teclas de letras volverán a sus funciones normales, enviando letras en minúsculas (CAPS corresponde a "capitals-mayúsculas").
BS	Cuando se pulsa esta tecla, desaparecerá de la pantalla el carácter situado inmediatamente a la izquierda de la posición corriente del cursor (BS es la abreviatura de "Backspace-retroceso").

SELECT — Esta tecla sólo tiene su propósito cuando usas un cartucho con programa. Al pulsarla, puedes hacer una SELECCION entre las diversas opciones que el programa ofrece.

RETURN — Esta tecla se usa para indicar la 'culminación' de cualquier instrucción o comando MSX-BASIC, o la terminación de un dato.

GRAPH — Cuando pulsas esta tecla, prácticamente todas las teclas producen desde ese momento algún carácter gráfico. Lo mismo se aplica cuando pulsas una de esas teclas simultáneamente con la tecla SHIFT. Pulsando de nuevo la tecla GRAPH todas las teclas vuelven a ejercer sus funciones normales. (GRAPH corresponde a "graphicos").

CODE — Cuando se pulsa esta tecla, se iluminará el piloto indicativo de CODIGO, y prácticamente todas las teclas producirán desde ese momento alguna letra específica y ligada al lenguaje (alguna 'clave'). Lo mismo se aplica cuando se pulsa una de las teclas simultáneamente con la tecla SHIFT. Pulsando de nuevo la tecla CODE, todas las teclas vuelven a ejercer su función normal.

Resumiendo

Prácticamente, cada una de las letras del 'home computer' es capaz de producir seis **caracteres** diferentes

1. Letras minúculas
2. Letras mayúsculas (+ SHIFT)
3. Símbolos gráficos 1, después de pulsar la tecla GRAPH
4. Símbolos gráficos 2 (+ SHIFT) después de pulsar la tecla GRAPH
5. Caracteres 1 específicos del lenguaje, después de pulsar la tecla CODE
6. Caracteres 2 específicos del lenguaje, (+ SHIFT) después de pulsar la tecla CODE.

Encontrarás una revisión completa en el Apéndice G de este manual.

MSX-BASIC

Capítulo 2

Trabajar con el MSX-BASIC

Capítulo 2.1

Cuando pones en marcha tu ordenador, después de haber sido instalado apropiadamente, verás que aparece en la pantalla:

```
MSX system
etc.
```

Luego, el ordenador comprobará si está inserto algún cartucho dentro de las ranuras. Si no lo está, automáticamente aparecerá en pantalla el siguiente texto:

```
MSX BASIC
etc.
```

El pequeño rectángulo, situado inmediatamente debajo de "OK", se denomina "cursor" e indica la posición en pantalla donde aparecerá el siguiente símbolo que teclees.

El número indica el número de celdillas o posiciones de memoria que todavía quedan libres en la memoria del ordenador.

El aviso "OK" aparecerá en la pantalla cada vez que el MSX-BASIC esté a **nivel de comando**, lo que significa que está 'preparado' para recibir tus instrucciones o comandos. En ese momento, tienes la posibilidad de usar el MSX-BASIC de dos modos diferentes: el 'Modo Directo' y el 'Modo Indirecto'.

El modo directo

Trabajando en el modo directo, se pueden usar prácticamente todas las 'frases' con verbos y funciones que componen el lenguaje MSX-BASIC, del que encontrarás una descripción completa en tu manual de referencia.

En el modo directo, todas las frases tecleadas serán interpretadas como **COMANDOS** y por tanto, serán ejecutadas inmediatamente después de culminar la frase pulsando la tecla RETURN. No quedan conservadas en la memoria del ordenador. Teclea el siguiente Ejemplo:

```
PRINT 5+2
```

Si has cometido un error al teclear, usa la tecla de retroceso (BS) para quitar el carácter situado inmediatamente a la izquierda del cursor. De esta manera puedes 'borrar' tantos símbolos como sean necesarios para hacer la corrección. Cuando esté correcta la línea de comando, pulsa la tecla RETURN.

Es importante que recuerdes pulsar la tecla RETURN para indicar al ordenador que has concluido de teclear un comando o una instrucción. Antes de que lo hagas, el ordenador meramente escucha y repica en la pantalla lo que estás tecleando, pero no pasa a ejecutar lo mandado hasta que se lo señalas mediante la pulsación de RETURN.

MSX-BASIC ejecutará inmediatamente el comando recibido y la respuesta aparecerá en la pantalla:

```
 7
OK
```

La divisa "OK" te dice que es MSX-BASIC y que ha vuelto al nivel de comando, esperando el siguiente comando o instrucción.

El modo indirecto

Trabajando en el modo indirecto, todas las frases permitidas en el lenguaje deben estar precedidas por un **número de línea**. La acción mandada en la frase, no será ejecutada inmediatamente después de pulsar la tecla RETURN, sino que quedará almacenada en la memoria del ordenador, en el lugar que correlativamente corresponde a ese número de línea. Por eso, se dice que le hemos dado una **INSTRUCCION** al ordenador.

Teclea ahora este ejemplo:

```
10 PRINT 5+2
```

Pulsa la tecla RETURN para indicar que has culminado la instrucción, y verás que nada sucede salvo que el cursor salta a la primera columna de la siguiente línea en espera de nuevas instrucciones. Esto significa también que ahora la línea 10 ha quedado depositada en la memoria del ordenador. Una **serie de instrucciones** como la del ejemplo, que enseñe al ordenador a llevar a cabo un determinado **trabajo**, es lo que se denomina **PROGRAMA**. Todas las instrucciones de ese programa, finalmente serán ejecutadas una a una y de acuerdo con su número de línea cuando des al ordenador el comando "RUN" para que 'EJECUTE EL PROGRAMA'.

Trabajando en el modo indirecto, debemos preceder cada frase con un número de línea y tienes multitud de números a elegir ya que puedes usar cualquiera entre 0 y 65529. Es aconsejable no usar números consecutivos cuando escribes un programa, sino más bien dejar saltos o intervalos entre ellos, ya que al hacerlo siempre podrás posteriormente añadir líneas que pasarán a colocarse en la posición correspondiente.

Puedes además pedir a MSX-BASIC que haga la numeración por ti, simplemente tecleando el comando "AUTO" antes de comenzar realmente con tus líneas de programa. Teclea el comando "AUTO" y verás que MSX-BASIC 'automáticamente' muestra 10 como tu primer número de línea. El asterisco después del 10 te indica que ya hay una línea con el número 10 en la memoria del ordenador. Cuando pulses la tecla RETURN, aparecerá automáticamente 20 como siguiente número, etc. Para detener el proceso automático de numeración, deberás pulsar simultáneamente las teclas CTRL y C.

MSX-BASIC permite el uso de 255 caracteres como máximo (incluyendo los espacios en blanco y los números de línea) por línea de programa. Una línea de programa puede contener varias frases **—comandos—** siempre que estén separadas por el signo dos puntos (:). Ensaya con un ejemplo:

```
10 PRINT 8+2:PRINT 6*4
RUN
 10
 24
OK
```

MSX-BASIC ha efectuado separadamente los comandos de cálculo y presentación que forman la instrucción número 10, que se denomina línea **multi-comando**.

Para quitar de la memoria del ordenador todas las líneas de programa previas, usa el comando "NEW" que deja la memoria como 'NUEVA'.

Diferencias entre las dos categorías de ACCIONES

En un sentido amplio, decimos que la frase tecleada representa una instrucción cuando la damos en el modo indirecto —precedida del número de línea— y que representa un comando cuando la damos en el modo directo —sin número de línea—. Ambas representan **acciones** que el ordenador ha de llevar a cabo posterior o inmediatamente, y deben empezar por una de las **palabras reservadas o palabras clave** del lenguaje MSX-BASIC.

Teclea el siguiente ejemplo de programa:

```
10 PRINT 3+1
20 LIST
30 PRINT 6-3
RUN
```

Dado que la acción LIST, que presenta en pantalla el LISTADO del programa almacenado en la memoria, es intrínsecamente una acción de la categoría **comando**, después de ser realizada la instrucción número 20 y presentar en pantalla el listado del programa, MSX-BASIC vuelve a nivel de comando y por tanto, NO ejecuta la instrucción de la línea 30.

En el manual de referencia MSX-BASIC se especifica claramente para cada una de las palabras clave que representan acciones, si pertenece a la categoría de comando o a la categoría de instrucción.

Diferencia entre acciones y funciones

Otras de las palabras clave del lenguaje MSX-BASIC representan **funciones**, y no pueden incluirse en una frase a no ser que vayan precedidas por una palabra clave que represente una **acción**. La palabra "FRE (0)" es un ejemplo de función, que se usa para pedir a MSX-BASIC que calcule cuántos octetos de memoria hay disponibles todavía. Ensaya el siguiente ejemplo.

```
NEW
10 FRE(0)
```

Ahora pulsa el comando RUN para que se ejecute el programa del ejemplo y verás que MSX-BASIC produce en pantalla un mensaje de error sintáctico:

```
Syntax error in 10
```

Es debido a que has usado una palabra admitida en el lenguaje, pero que representa una **función** sin preceder la frase de una palabra que represente una acción. Ensaya ahora el siguiente ejemplo:

```
10 PRINT FRE(0)
RUN
```

Ahora MSX-BASIC realiza la acción de EXPONER en pantalla el resultado de la función que halla la cantidad de memoria LIBRE (FRE) que hay.

Tecleando instrucciones

La palabra PRINT es una de las acciones —de categoría instrucción— que utilizarás muy a menudo para hacer que MSX-BASIC EXPONGA en pantalla datos y mensajes. Para ahorrarte tiempo y problemas, puedes usar en lugar de la palabra PRINT, el signo de interrogación (?), que actúa en ese momento como si fuera una palabra clave, además será transformado en dicha palabra PRINT. Veamos cómo funciona. Teclea:

```
10 ? 2+9
```

Termina la instrucción pulsando RETURN y teclea el comando LIST para que LISTE el programa. Verás:

```
10 ? 2+9
LIST
10 PRINT 2+9
```

MSX-BASIC te permite teclear claves e incluso frases, pulsando únicamente una tecla ('mono-tecla'). Las teclas definibles o funcionales F1 a F10, han sido previstas para esa finalidad. En la parte inferior de la pantalla, verás las asignaciones que tienen previstas las teclas F1 a F5. Cuando pulses la tecla SHIFT simultáneamente con una de ellas, verás en la pantalla los comandos e instrucciones que corresponden a las teclas F6 a F10.

También puedes pedir a MSX-BASIC que produzca un listado con las claves y frases que han sido definidas para estas teclas funcionales; tecleando el comando KEY LIST (key en este caso es TECLA).

Puedes también alterar las claves o frases asociadas a estas teclas definibles, usando la clave e instrucción "KEY".

```
KEY 1,"GOSUB"
```

Y a partir de este momento, la tecla definible F1 producirá al ser pulsada la clave de instrucción "GOSUB".

Puedes quitar de la pantalla las referencias a las teclas definibles, usando la frase "KEY OFF" y hacer que vuelvan a aparecer usando la frase "KEY ON".

Cuándo y cómo usar los signos " " : ; . ,

MSX-BASIC ejecutará perfectamente todo lo que se le encargue, siempre que lo teclees correctamente. Y eso se aplica también a los diversos signos de puntuación. Todos ellos tienen un propósito específico cuando se emplean para escribir un programa.

Todos los caracteres colocados dentro de **comillas** serán producidos en pantalla exactamente tal y como los teclees. Puedes verlo fácilmente si tecleas estos dos ejemplos:

```
10 PRINT 5+2
20 PRINT "5+2"
RUN
```

Con la primera instrucción, MSX-BASIC efectúa el cálculo 5+2=7. Pero en la segunda instrucción reproduce **LITERALMENTE** el texto que has incluido dentro de las comillas, i.e. 5+2. De ahí que se diga que es un dato de índole **literal** o alfanumérico.

El signo de puntuación **dos puntos** (:) sirve para separar comandos dentro de una línea, como ya hemos mencionado.

El signo de puntuación **punto y coma** (;) puede usarse en una instrucción o en un comando "PRINT", para hacer que el cursor después de exponer un dato se coloque en la posición inmediatamente a continuación en la pantalla. Si se coloca al final de una línea, impide que el cursor pase a la línea siguiente.

También puede usarse en combinación con la instrucción "INPUT", cuando se incluye una pregunta a exponer antes de realizar la imposición de datos.

El signo de puntuación **punto** (.) puede usarse en unión de las claves de comandos "LIST", "AUTO" y "DELETE", cuando se quiere hacer referencia a la última línea de programa que se haya estado tratando.

El signo de puntuación **coma** (,) sólo puede incluirse dentro de los comandos e instrucciones, tal y como se escribe en el manual de referencia MSX-BASIC.

Mensajes de error

Cuando MSX-BASIC encuentra un error en la ortografía de las palabras clave, o en las sintaxis de los comandos e instrucciones producirá en pantalla un mensaje explicando el error detectado. Encontrarás un resumen completo de dichos mensajes en el Apéndice A de este manual.

Cambio de un programa

Capítulo 2.2

Cualquier **línea de programa** puede fácilmente cambiarse por completo y en cualquier momento, tecleando una nueva instrucción precedida con el mismo número de línea.

Se pueden añadir nuevas líneas a un programa dándoles números de línea que no hayan sido empleados todavía y recordando siempre que has de escoger un número que coloque a la nueva línea en la secuencia apropiada.

Las líneas de programa pueden borrarse usando la clave de comando "DELETE". Si quieres quitar una sola línea, teclea a continuación de la clave simplemente el número de línea que deseas quitar. También es posible cambiar líneas de programa cuando aparecen listadas en pantalla. Es lo que se denomina 'edición de plana' (y también edición de pantalla).

Edición de plana

La edición de plana significa que las líneas de programa que aparecen en pantalla pueden cambiarse usando las teclas de movimiento del cursor y las teclas marcadas INS, BS y DEL.

Ensaya con el siguiente ejemplo. Teclea:

```
NEW
10 pontt
```

Lo que realmente querías usar era "PRINT", y obviamente has cometido un error de tecleado. No te preocupes si has usado minúsculas, MSX-BASIC convertirá todas las palabras clave del lenguaje a letras mayúsculas.

```
LIST
```

Teclea ahora el comando para listar el programa, y verás en pantalla:

```
10 PONTT
```

Dado que quieres enmendar el error cometido, tendrás que cambiar esta línea de programa. Primero sitúa el cursor sobre la letra "O" de la palabra "PONTT" usando las teclas de desplazamiento del cursor. Ahora pulsa la tecla INS para pasar al modo de inserción, y observarás que el cursor se hace más estrecho. Eso significa que cada símbolo que teclees a partir de ahora, se insertará entre los símbolos que aparecen en pantalla. Ahora pulsa la tecla marcada "R" y verás que la letra "R" se sitúa entre la "P" y la "O". Pulsa de nuevo la tecla de inserción y verás que el cursor retorna a su tamaño normal. Eso significa que cada símbolo que teclees ahora SUSTITUIRA al símbolo que aparece en pantalla en la posición del cursor. Por lo tanto, pulsa la tecla "I" y sustituirá a la letra "O" para que obtengas:

```
10 PRINTT
```

El cursor está situado sobre la "N". Lo que hay que hacer ahora es SUPRIMIR una de las "T". Por lo tanto, desplaza el cursor hasta situarlo sobre la primera y pulsa la tecla DEL para poder 'delectar' —o sea, suprimir— dicha letra. Al pulsarla, verás que desaparece la primera "T" y que la otra se ha corrido una posición a la izquierda para rellenar el hueco producido.

Observa que los cambios solamente son efectuados por MSX-BASIC cuando pulsas la tecla RETURN, que como siempre indica la culminación de un comando, instrucción, operación, etc.

Cuando teclees ahora el comando "LIST", el 'editor' de MSX-BASIC te mostrará que la línea ha quedado en la forma:

```
10 PRINT
```

que es exactamente lo que querías.

Asegúrate siempre que los nuevos comandos o las nuevas instrucciones después de una edición se introducen usando una línea vacía de pantalla. Supongamos, por ejemplo, que tienes en pantalla el siguiente texto:

```
10 PRINT 9+2
20 PRINT 3-6
```

Ahora supongamos que el cursor está situado sobre el "2" de la línea de programa 20, dado que acabas de cambiar la línea 10. Si ahora tecleas la palabra clave "LIST", obtendrás este resultado:

```
10 PRINT 9+2
LISTRINT 3-6
```

Ahora, al culminar el comando "LIST" usando la tecla "RETURN", verás que MSX-BASIC te da un mensaje de error dado que el comando "LISTRINT 3-6" no existe en el lenguaje MSX-BASIC. Desde luego, puedes evitar eso moviendo el cursor a una línea de pantalla vacía, o pulsando simultáneamente las teclas "SHIFT" y "HOME" antes de haber tecleado el comando "LIST".

Si el contenido de una línea en particular no tiene utilización posterior, puedes pulsar simultáneamente la tecla "CTRL" y la tecla "E", con lo que quitarás todos los símbolos que aparecen en esa línea, desde la posición presente del cursor y hasta el final de la línea.

Tecla "STOP"

La tecla "STOP" no ejerce ninguna función cuando MSX-BASIC está a nivel de comando.

Cuando MSX-BASIC está ejecutando un programa, puedes hacer que PARE si pulsas la tecla "STOP". Al mismo tiempo reaparecerá en pantalla el cursor. El programa continuará cuando pulses otra vez la tecla "STOP".

Si quieres interrumpir un programa y hacer que MSX-BASIC se coloque a nivel de comando, pulsa simultáneamente las teclas "CTRL" y "STOP". Verás que en pantalla aparece el siguiente mensaje:

```
Break in nnnnn
```

Para indicar que se ha 'interrumpido' la ejecución en el número de línea designado por "nnnnn".

Constantes y variables

Capítulo 2.3 La memoria de tu ordenador MSX está capacitada para almacenar otra información además de las líneas de programa con las claves de los comandos y de las funciones. Hay otros elementos que juegan un papel importantísimo durante la ejecución de un programa. Puede considerarse que pertenecen a dos categorías: **constantes** y **variables**.

Constantes

Las constantes representan información que no cambia durante la ejecución del programa. Hay dos clases diferentes de constantes: **numéricas y alfanuméricas**. Una constante alfanumérica está representada por una cadena de caracteres con una longitud máxima de 255 caracteres, y están encerrados entre **comillas** ("...").

Ejemplo: "HOLA" o "12345".

Incluso, aunque la constante "12345" incluye cifras, **no** se considera de índole numérica y no puede intervenir en cálculos matemáticos.

Una constante numérica, un **numeral**, es de hecho, una cifra positiva o negativa expresada en algún sistema de numeración. La coma decimal que separa la parte entera y fraccionaria, debe ser sustituida por el punto decimal en MSX-BASIC. Ejemplo: 123,45 se escribe 123.45.

MSX-BASIC reconoce seis clases diferentes de constantes numéricas:

1. **Constante entera**
   Es cualquier número en base 10 entre —32768 y +32767
   Ejemplo: 13 ó —25
2. **Constante de punto fijo**
   Es un número con cifras en la parte fraccionaria
   Ejemplo: 15.73 ó —178.3
3. **Constante de punto flotante**
   Es un número escrito según la notación **científica**
   Ejemplo: 2.34E8 (=2.34*10^8=234,000,000)
4. **Constante hexadecimal**
   Es un número usando base 16. Está precedido por "&H"
   Ejemplo: &H3F
5. **Constante octal**
   Es un número expresado en base 8. Está precedido por "&O"
   Ejemplo: &O37
6. **Constante binaria**
   Es un número expresado en base 2. Está precedido por "&B"
   Ejemplo: &B01010111

Variables

Una variable es un sitio de la memoria identificado por un **nombre** y preparado para almacenar una pieza de información —**dato**— usada por el ordenador. Cada variable tiene un nombre **unívoco**, dado por el programador, que permite al ordenador encontrar el **dato** alojado en ese lugar. Hay dos clases diferentes de variables: **numéricas y alfanuméricas**.

El nombre de una variable alfanumérica, termina siempre con el signo dólar ($).
Ejemplo: A$="HOLA" o AB$="12345".

Una variable **numérica** pude contener, tener como valor, cualquier numeral.

Una variable numérica entera sólo puede contener números enteros entre —32768 y 32767. El nombre de tal variable siempre termina con el símbolo porcentaje (%).
Ejemplo: A%=10.

Simple y doble precisión

Las constantes numéricas y los valores de las variables numéricas pueden darse en simple o en doble precisión:

En **simple precisión**, los numerales ocupan 4 octetos de memoria y en notación normal, constan de un máximo de 6 cifras o dígitos.

Una constante de punto fijo en simple precisión, queda designada colocando al final un signo de exclamación (!).
Ejemplo: 12.5!

Una constante de punto flotante en simple precisión, queda designada incluyendo la letra "E" para indicar el exponente.
Ejemplo: 2.34E8

Una variable numérica de simple precisión tiene como último carácter del nombre que la identifica, el signo de exclamación (!).
Ejemplo: AB!

En **doble precisión**, los numerales están ocupando 8 octetos de espacio en memoria, y pueden expresarse en forma normal, con un máximo de 14 cifras o dígitos.

Una constante de punto fijo en doble precisión está indicada colocando al final el símbolo diesi "#" (almohadilla).
Ejemplo: 23.33#

Una constante de punto flotante en doble precisión, queda indicada incluyendo la letra "D" como símbolo del exponente.
Ejemplo: 3.768D-7

Una variable numérica en doble precisión, está designada al usar el símbolo "#" como último carácter del nombre que la identifica.
Ejemplo: DA#

Si un nombre de variable no tiene como último carácter uno de los signos "$", "%", "!" o "#", el MSX-BASIC considerará que esa variable es numérica y de doble precisión.

Ejemplos:	ME#	es una variable de doble precisión, ocupando 8 octetos de memoria.
	PT%	es una variable numérica entera, ocupando 2 octetos.
	ZS!	es una variable numérica de simple precisión ocupando 4 octetos.
	ME	es una variabie numérica de doble precisión, ocupando 8 octetos.
	AB$	es una variable alfanumérica, ocupando en memoria 3 octetos además de un espacio con un número de octetos igual al número de caracteres reales del valor literal que contiene. (Los incluidos en la cadena, dentro de las comillas).

MSX-BASIC tiene prescrito un espacio máximo de 200 octetos para alojar los literales correspondientes a las variables alfanúmericas usadas en un programa; y ese máximo puede cambiarse usando la clave de instrucción "CLEAR".

Con MSX-BASIC, la clase de variable puede determinarse además usando las claves de instrucciones "DEFINT", "DEFSTR", "DEFSNG" y "DEFDBL", que corresponden respectivamente a variables enteras, alfanuméricas, ('cadenas'), simple y doble precisión. Encontrarás detalles completos en el manual de referencia.

Conversión de clases de variables

MSX-BASIC convertirá, si es necesario, los valores de las variables numéricas de una a otra clase.
Ejemplo 1:

```
NEW
10 A%=13.03
20 PRINT A%
RUN
 13
```

En este ejemplo un numeral de punto fijo se asigna como valor a una variable entera. Las cifras fraccionarias, después del punto decimal, se eliminan.
Ejemplo 2:

```
NEW
10 E#=6!/7!
20 PRINT E#
RUN
 .85714285714286
```

Incluso aunque el cálculo 6 dividido 7 está indicado con numerales de simple precisión, el resultado de la división está especificado en doble precisión, dado que el nombre de la variable "E#" así lo requiere.
Ejemplo 3:

```
NEW
10 E!=6/7
20 PRINT E!
RUN
 .857143
```

Incluso aunque el cálculo "6/7" se exige con operandos en doble precisión (sin ningún signo) el resultado se obtiene en simple precisión tal y como lo requiere el nombre de la variable utilizada "E!".
Ejemplo 4:

```
NEW
10 B!=SQR(2)
20 A=B!
30 PRINT B!;A
RUN
 1.41421  1.41421
```

A pesar de que la variable "A" está definida como de doble precisión, sólo aparecen 6 cifras al exponer su valor en pantalla, dado que el contenido de la variable de doble precisión (A) se ha hecho idéntico al de la variable de simple precisión (B!), que contiene el resultado de la raíz cuadrada de dos.

La conversión de numerales también puede conseguirse mediante las funciones representadas por las claves "CINT", "CDBL" y "CSNG", para expresar el numeral como entero, doble y simple precisión respectivamente. Además, la **índole** del valor de una variable puede cambiarse de **numeral** a **alfanumérica**, usando la función expresada por la clave "STR $". Los detalles completos en relación con estas funciones, pueden verse en el manual de referencia.

Nombres de variables

El programador puede elegir por sí mismo los nombres de las variables usadas en sus programas, siempre y cuando se cumplan las siguientes condiciones:

1. El nombre debe comenzar siempre con una **letra** (de la A a la Z), mientras que los otros caracteres pueden ser bien letras o cifras.
2. Incluso aunque el nombre puede constar de más de **dos caracteres**, MSX-BASIC sólo reconoce los dos primeros del nombre. Por tanto, los nombres "PO" y "POST" hacen referencia a una sola y a la misma variable.
3. El nombre de la variable no puede coincidir con ninguna de las palabras reservadas **(claves)** enunciadas en el Apéndice F de este manual.

Contenido de las variables: valores

Se puede dotar de un cierto valor a una variable usando la clave de instrucción "LET", para hacer que HAGA el valor de la variable igual a un determinado numeral o literal. Ejemplo:

```
10 LET A=10
20 LET B$="HOLA"
```

El uso de la palabra "LET" no es estrictamente necesario y se consigue el mismo efecto sin usarla. Pero las frases del lenguaje siempre empiezan por una palabra de **acción**.

También puede IMPONERSE un valor como contenido de una variable usando la clave de instrucción "INPUT". Cuando el ordenador encuentra dicha clave, esperará a que se le introduzca una determinada información por el teclado, que culminada mediante la pulsación de la tecla "RETURN", será alojada en el sitio señalado por el nombre de variable que se menciona en la instrucción "INPUT". También puede añadirse una pregunta o aviso sobre el dato que se está esperando, al usar la instrucción "INPUT".
Ejemplo:

```
NEW
10 INPUT "DIME UN NUMERO";A
20 INPUT "DIME TU NOMBRE";B$
RUN
DIME UN NUMERO?
```

El **signo de interrogación** se añade automáticamente.
En la línea 10 de programa, MSX-BASIC aceptará información numérica. En la línea 20 de programa, cualquier información será considerada de índole alfanumérica, aunque se pulsen cifras.

Variables especiales

Con MSX-BASIC puedes usar ciertas variables especiales, con nombres clave, al escribir tus programas. Esas variables tienen por nombres: "TIME", "VDP", "BASE" y "SPRITE$".
Ejemplo:

```
NEW
10 PRINT "INICIO":TIME=0
20 FOR I%=1 TO 1000:NEXT I%
30 PRINT "HAN TRANSCURRIDO";TIME/50;"SEGUNDOS"
RUN
INICIO
HAN TRANSCURRIDO .8 SEGUNDOS
```

El ejemplo anterior muestra que el ordenador tarda .8 segundos en ejecutar la instrucción "FOR...NEXT" de la línea 20 del programa.

La variable especial "SPRITE$" se explica con detalle en el capítulo 2.12. Las variables de nombres "VDP" y "BASE", solamente deben ser usadas por aquéllos que están profundamente familiarizados con el funcionamiento del procesador de video.

Calculos

Capítulo 2.4 MSX-BASIC reconoce cinco operaciones aritméticas cuyo resultado es un numeral:

1. Adición (+)
2. Sustracción (—)
3. Multiplicación (*)
4. División (/)
5. Potenciación (^)

Estas operaciones pueden combinarse formando **expresiones** como por ejemplo:

```
NEW
10 PRINT 40/5+3
RUN
 11
```

Recuerda por favor, que las operaciones aritméticas se efectúan según el orden de **prioridad**:

1. Potenciación
2. Multiplicación y división
3. Adición y/o sustracción

Este orden puede, por supuesto, alterarse usando **paréntesis** como en el siguiente ejemplo:

```
NEW
10 PRINT 40/(5+3)
RUN
 5
```

División entera MSX-BASIC es capaz de efectuar la división entera **congruente,** lo que significa que la división será efectuada sólo con números enteros y que se dará el resultado también como un número entero. El signo de la división entera es la barra invertida (\). Primero, ensaya una división real:

```
NEW
10 PRINT 10/4
RUN
 2.5
```

Y observa la diferencia cuando utilizas división entera:

```
NEW
10 PRINT 10\4
RUN
 2
```

El dividendo y el divisor son convertidos a numerales enteros antes de efectuar la división entera.

Cálculos de congruencias ('Restos Módulos')

El cálculo de congruencias trabaja con los restos de la divisiones enteras. Y el signo de la operación correspondiente está caracterizado por la palabara clave "MOD", que corresponde a módulo. Por ejemplo:

```
NEW
10 PRINT 25.68 MOD 6.99
RUN
 1
```

En este ejemplo, los numerales 25.68 y 6.99 son considerados como enteros (25 y 6) antes de que se efectúe la división. Se diría "resto de 25 módulo 6" y el resultado es el valor residual —**resto**— de la división, que en este caso es 1.

División por cero y rebase de tamaño

Si se efectúa una división por cero, MSX-BASIC informará del hecho con un mensaje de error y detendrá la ejecución del programa.
Si el campo receptor del resultado de una división aritmética es demasiado corto para contener el resultado de un cálculo, MSX-BASIC también producirá un mensaje de error y detendrá la ejecución sucesiva del programa.

Variables

Si se menciona el nombre de una variable dentro de una expresión matemática, se ejecuta la operación, tomando el contenido —el **valor**— de esa variable. Por ejemplo:

```
NEW
10 A=10:B=5
20 C=A*B
30 PRINT C
RUN
 50
```

En la línea 20 de programa, se multiplica el numeral 10 (el contenido de la variable A), por el número 5 (el contenido de la variable B) y se asigna como valor de la variable C.

No pueden efectuarse operaciones aritméticas con datos de índole alfanumérica. Es posible, sin embargo, 'empalmar' los alfanuméricos mencionándolos directamente como constantes o como contenidos de variables alfanuméricas. Ejemplo:

```
NEW
10 B$="MSX ":C$="- COMPUTER"
20 A$=B$+C$
30 PRINT A$
40 PRINT B$+"- BASIC"
RUN
MSX - COMPUTER
MSX - BASIC
```

En este caso, el contenido de la variable alfanumérica "A$" es "MSX-COMPUTER".

Funciones matemáticas

Algunas de las fórmulas más frecuentemente usadas en matemáticas están representadas en MSX-BASIC por palabras clave que designan **funciones**.

La función "ABS" obtiene el valor **absoluto** de un número.
La función "COS" calcula el **coseno** de un número representando la medida de un ángulo en radianes.
La función "EXP" calcula el **antilogaritmo** de un número (el número **e=2.71** elevado a dicho número).
La función "LOG" da el **logaritmo natural** del número mencionado como argumento.
La función "SIN" calcula el **seno** de un ángulo expresado en radianes.
La función "SQR" calcula la **raíz cuadrada** del argumento.
La función "TAN" calcula la **tangente** del ángulo en radianes que aparece como argumento.
Ensaya este ejemplo:

```
NEW
10 X=25
20 PRINT X;SQR(X)
RUN
 25  5
```

Otras funciones matemáticas pueden componerse mediante expresiones numerales a partir de las funciones intrínsecas del lenguaje. Además, puedes definir otras funciones matemáticas por ti mismo, usando la clave de instrucción "DEF FN". Observa el siguiente ejemplo:

```
NEW
10 DEF FNAB(X,Y)=X^3/Y^2
20 X=6:Y=2
30 T=FNAB(X,Y)
40 PRINT T
RUN
 54
```

Las variables cuyos nombres X e Y se mencionan en la línea 10 del programa que define la función, reciben el nombre de **argumentos formales** o nominales de la función. En la línea 20 de programa se dota a estas variables de un valor numérico elegido. En la línea 30 del programa, al **citar** la función previamente definida, los valores que intervienen (6 y 2) reciben el nombre de **argumentos actuales,** o efectivos. La variable cuyo nombre es "T" contiene el resultado de operar la función con los argumentos actuales, y por tanto es 6^3/2^2.

MSX-BASIC y la lecto-grabadora de datos

Capítulo 2.5

MSX-BASIC permite el uso de un equipo de cinta magnética en cassette para depositar información, y recuperarla posteriormente, en forma de programas y ficheros.

Para guardar programas en el cassette

Un programa puede trasvasarse de la memoria al cassette. usando la clave de comando "CSAVE". El procedimiento es el siguiente:

1. Coloca una cinta en la lecto-grabadora de cassette, y rebobínala completamente.
2. Bobina el cassette avanzando hasta que hayas pasado el trozo de cinta metálica de la cabecera.
3. Oprime simultáneamente los botones PLAY y REC de la lecto-grabadora, colocándola en el modo de grabación.
4. Teclea la clave de comando "CSAVE" inmediatamente seguida del nombre con el que quieres identificar el programa que GUARDE el ordenador en el cassette. Encierra el nombre entre comillas (CSAVE "nombre de programa").
5. Culmina el comando anterior, pulsando la tecla RETURN.

El programa será transferido de la memoria al cassette. Hemos hecho que el ordenador GUARDE el programa.

Para cargar un programa del cassette

Puede trasvasarse un programa del cassette a la memoria del ordenador, usando la clave de comando "CLOAD". El procedimiento es como sigue:

1. Inserta el cassette que contiene dicho programa en la lecto-grabadora y rebobina completamente.
2. Oprime el botón PLAY para colocarlo en el modo de reproducción ('lectura').
3. Teclea el comando: CLOAD "nombre de programa".
4. Culmina el comando anterior pulsando la tecla RETURN.

Si no se menciona ningún nombre de programa a continuación de la clave de comando "CLOAD", se transferirá a la memoria del ordenador el primer programa que se encuentre en la cinta. Cuando al cargar un programa con un determinado nombre, MSX-BASIC encuentra programas con nombre diferente, indicará en la pantalla que se lo 'SALTA' mediante el mensaje:

```
SKIP (NOMBRE DE PROGRAMA)
```

Y continuará la búsqueda en la cinta del programa exigido. Indicará que lo ha encontrado mediante el mensaje:

```
FOUND (NOMBRE DE PROGRAMA)
```

Al concluir esa tarea, el MSX-BASIC volverá al nivel de comando. Esta tarea se denomina CARGA DE UN PROGRAMA.

Depositando datos en el cassette

La información recopilada mientras se está ejecutando un programa, puede también ser guardada en cassette. Los trozos de información se van depositando sucesiva y correlativamente en la cinta, formando una colección de datos denominada **fichero;** y en este caso, de la categoría **fichero secuencial,** ya que los datos están depositados en la cinta según la **secuencia** en que se han generado en el programa. Todo fichero recibe un nombre que lo identifica y tiene que ser preparado **—ABIERTO—** para poder usar en un programa la información que contiene.

El número de ficheros admitido dentro de un programa, tiene que especificarse mediante la clave de instrucción "MAXFILES". Prueba el ejemplo siguiente:

```
5 MAXFILES=1
10 OPEN "CAS:DIRE" FOR OUTPUT AS#1
20 A=15:B=23.5:C=10:D=25.2:E=13
30 PRINT #1,A;B;C
40 PRINT #1,D;E
50 CLOSE #1
```

En este ejemplo, en la línea 10, el fichero llamado "DIRE" es ABIERTO como fichero de SALIDA, y queda identificado en el programa como fichero #1.

La palabra clave "CAS:" indica que el fichero está contenido en la unidad de almacenamiento CASSETTE.

La información realmente queda depositada en el cassette usando la clave de instrucción "PRINT #". Después de cada instrucción "PRINT #", el sistema genera y graba en la cinta el código correspondiente a la marca de "retorno de carro" "CR" y el correspondiente a la marca "avance de línea" "LF". Además, y automáticamente, se coloca el código correspondiente al signo de puntuación como para separar las variables numéricas mencionadas en la instrucción. En nuestro ejemplo anterior, si pudieramos examinar la colocación de los datos en la cinta, veríamos la siguiente 'imagen de fichero':

15	,	23.5	,	10	CR	LF	25.2	,	13	CR	LF

Al 'exponer' variables alfanuméricas para que sea grabado su contenido en la cinta, DEBE añadirse explícitamente la coma que delimita cada uno de los datos dentro de la instrucción "PRINT #"...; como muestra el siguiente ejemplo:

```
5 MAXFILES =1
10 OPEN "CAS:NOME" FOR OUTPUT AS#1
20 A$="JOHN":B$="PETE"
30 PRINT #1,A$;",";B$
40 CLOSE #1
```

Eso producirá la siguiente 'imagen de fichero' en la cinta:

JOHN	,	PETE	CR	LF	

Asegúrate que los puntos 1 a 3 del procedimiento, explicado para guardar programas, también se ejecutan antes de intentar transferir datos al cassette.

Recuperando datos del cassette

La información depositada en el cassette puede ser recuperada y trasvasada a la memoria del ordenador por medio de la clave de instrucción "IMPUT #".
Para imponer los valores a las variables numéricas de nuestro ejemplo anterior, deberá usarse la siguiente serie de instrucciones:

```
5 MAXFILES=1
10 OPEN "CAS:DIRE" FOR INPUT AS#1
20 INPUT #1;V,W,X
30 INPUT #1;Y,Z
40 CLOSE #1
```

Para imponer los valores a la variables alfanuméricas anteriores, necesitas usar por ejemplo, las siguientes instrucciones:

```
5 MAXFILES=1
10 OPEN "CAS:NOME" FOR INPUT AS#1
20 INPUT #1,Y$,Z$
30 CLOSE #1
```

Una instrucción "IMPUT #"... impone valores en las variables usando toda la información que aparece hasta encontrar la primera coma, o la pareja de código "CR" y "LF".
Si usáramos la instrucción "LINE INPUT #"..., se inpondría como valor de la variable toda la información que aparece hasta encontrar la pareja de códigos "CR" y "LF", como en el siguiente ejemplo:

```
NEW
5 MAXFILES=1
10 OPEN "CAS:NOME" FOR INPUT AS#1
20 LINE INPUT #1,X$
30 CLOSE #1
```

En este ejemplo, el contenido de X$ sería "JOHN, PETE". Con la clave de función "EOF" puedes hacer que el sistema te comunique cuándo ha llegado al FINAL DE FICHERO.

Asegúrate siempre que sigues los puntos 1 a 2 del procedimiento detallado para cargar un programa del cassette a la memoria del 'home computer'.

Un fichero de 'señas'

El siguiente programa puede usarse para apuntar y consultar nombres, direcciones y teléfonos. La información se conserva en cassette y puede consultarse en cualquier momento.

```
NEW
10 SCREEN 0:KEY OFF:WIDTH 39:MAXFILES=1
20 CLS:LOCATE 10,2:PRINT "MENU"
30 ON KEY GOSUB 100,400,700
40 LOCATE 5,5:PRINT "F1=IMPOSICION DE DIREcciones"
50 LOCATE 5,7:PRINT "F2=EXPOSICION DE DIREcciones"
60 LOCATE 5,9:PRINT "F3=CONCLUSION DE TRABAJO"
70 KEY(1) ON:KEY(2) ON:KEY(3) ON
80 GOTO 80
100 CLS
110 PRINT:PRINT "prepara tu lecto-grabadora y pulsa
    simultáneamente las teclas PLAY y REC"
120 PRINT:PRINT:INPUT "PREPARADO (S/N)";F$
130 IF F$="N" GOTO 120
140 IF F$="n" GOTO 120
150 OPEN "CAS:DIRE" FOR OUTPUT AS#1
160 CLS
170 LOCATE 1,5:INPUT "NOMBRE";A$
180 LOCATE 1,7:INPUT "CALLE";B$
190 LOCATE 1,9:INPUT "CIUDAD";C$
200 LOCATE 1,11:INPUT "TELEFONO";D$
210 LOCATE 1,20:INPUT "CONFORME (S/N)";F$
220 IF F$="N" GOTO 160
230 IF F$="n" GOTO 160
240 PRINT #1,A$;",";B$;",";C$;",";D$
250 LOCATE 1,22:INPUT "OTRA DIRECCION (S/N)";F$
260 IF F$="N" GOTO 290
270 IF F$="n" GOTO 290
280 GOTO 160
290 CLOSE #1:RETURN 20
400 CLS
410 PRINT:PRINT "prepara tu lecto-grabadora y oprime
    la tecla PLAY"
420 PRINT:PRINT:INPUT "CONFORME (S/N)";F$
430 IF F$="N" GOTO 420
440 IF F$="n" GOTO 420
450 OPEN "CAS:DIRE" FOR INPUT AS#1
460 CLS:PRINT
470 IF EOF(1) THEN 570
480 INPUT #1,A$,B$,C$,D$
490 PRINT "NOMBRE    :";A$:PRINT
500 PRINT "CALLE     :";B$:PRINT
510 PRINT "CIUDAD    :";C$:PRINT
520 PRINT "TELEFONO  :";D$
530 LOCATE 1,22:INPUT "OTRA DIRECCION (S/N)";F$
540 IF F$="N" GOTO 570
550 IF F$="n" GOTO 570
560 GOTO 460
570 CLOSE #1:RETURN 20
700 CLS:PRINT "CONCLUSION DE TRABAJO"
710 END
```

Ejecuta el programa y ensáyalo imponiendo unas cuantas direcciones. Quedarán registradas en cinta cuando pulses la tecla F1 y serán recogidas de la cinta y transferidas a la memoria cuando pulses la tecla F2. No te olvides rebobinar previamente el cassette antes de intentar recuperar las direcciones y examinarlas en pantalla.

Si el programa no funciona adecuadamente, comprueba lo que has tecleado porque se suelen cometer errores.

Otras de las claves de instrucciones para depositar y recuperar ficheros en el cassette, son: BLOAD, BSAVE, LOAD "CAS", SAVE "CAS", MERGE, MOTOR. Consulta el manual de referencia.

Para CARGAR programas registrados en cintas comerciales, sigue cuidadosamente las instrucciones que acompañan a dichos cassettes.

TABLAS Y SERIES

Capítulo 2.6

Cuando en un programa se usan datos constantes que guardan mucha relación entre sí, es conveniente agruparlos para formar series y tablas.

Series de constantes

Para hacer una **serie** de datos, se usa la clave de instrucción "DATA". Los datos pueden ser 'apuntados' como valores de variables usando la instrucción "READ", como en el siguiente ejemplo:

```
NEW
10 READ X$:PRINT X$
20 GOTO 10
30 DATA ENE,FEB,MAR
RUN
ENE
FEB
MAR
Out of DATA in 10
```

Observa que las partidas de información mencionadas en la instrucción "DATA" están separadas por comas. La instrucción "READ" hace que el ordenador APUNTE la siguiente pieza de información de la serie, como valor de la variable mencionada en la instrucción. Internamente, el sistema mantiene un "puntero" para saber el número de datos de la serie que ya lleva apuntados. La instrucción "GOTO 10" que aparece en la línea 20, obliga a que el ordenador VAYA a la línea 10 con lo que vuelve a repetirse dicha instrucción. El mensaje de error "Out of DATA in 10" está provocado por el hecho de no disponer de más datos en la serie, cuando se intenta ejecutar por cuarta vez la instrucción "READ".

La clave de instrucción "RESTORE" hace que el ordenador RESTAURE el puntero al primer elemento de la primera de las series que aparezcan en el programa. Añade la siguiente instrucción al programa anterior y ejecútalo ahora.

```
15 RESTORE
```

El programa continúa ininterrumpidamente repitiendo el primer dato de la serie "ENE", hasta que hagas que PARE la ejecución pulsando simultáneamente las teclas CTRL y STOP. Usa el comando "NEW" para dejar la memoria como NUEVA.

Tablas de variables

También puedes agrupar variables de una forma **regular** para formar una TABLA. Dado que las variables representan sitios de la memoria, para definir una tabla es preciso indicar al ordenador que 'ocupe' un determinado espacio en la memoria para alojar los valores de las variables; y se hace mediante la clave de instrucción "DIM", ya que el espacio a ocupar depende de las DIMENSIONES de la tabla. Ejemplo:

```
10 DIM A(20)
```

Significa que se reservan 21 (0 a 20) espacios contiguos de la memoria, preparados para valores numéricos y cuyo nombre genérico viene designado por la letra "A". Cuando quieras usar una de las variables de la tabla en una línea de programa, ese nombre no es obviamente suficiente para designar los 21 datos diferentes. Necesitas especificar a cuál de las 21 variables estás haciendo referencia. Eso debe hacerse añadiendo un **sufijo,** que es un número colocado entre paréntesis al final del nombre genérico de la variable.

```
20 PRINT A(3)
```

Hace referencia a la variable que ocupa el **cuarto lugar** de la tabla. En lugar de un número para el sufijo, puedes usar evidentemente el nombre de otra variable, cuyo valor sería el tomado para saber a la que te estás refiriendo.

```
NEW
10 DIM A(18)
20 A(3)=15:B=0
30 PRINT A(B)
RUN
 0
```

La respuesta que MSX-BASIC da es 0 porque el contenido del sitio de la memoria designado A (0) no ha sido dotado de ningún valor. Pero si cambias la línea 20 del programa para que sea:

```
20 A(3)=15:B=3
```

Y ejecutas de nuevo el programa, MSX-BASIC te dará como respuesta "15", que es el dato contenido en el sitio de memoria llamado A(3), que a su vez, repetimos, es la **cuarta** de las variables agrupadas en la tabla.

Tablas de dos dimensiones

También es posible hacer que ocupe espacios 'rectangulares' para tablas de dos dimensiones, con la misma clave de instrucción "DIM", pero con dos indicadores de dimensión. Por ejemplo:

```
10 DIM A(1,3)
```

Eso reservaría en memoria un área similar a la siguiente:

	0	1	2	3
0				
1				

Ahora, para referirse a una de las ocho variables numéricas agrupadas en la tabla A, debemos mencionar el sufijo de dos dimensiones correspondiente, v.g.:

```
20 A(1,2)=64
```

Esta instrucción alojaría el número 64 en el sitio de la memoria que ocupa la línea 1, columna 2 del área que ha sido reservada:

	0	1	2	3
0				
1			64	

Para todas las variables numéricas, MSX-BASIC reservará 11 sitios contiguos de memoria (0 a 10), incluso sin mencionar previamente la instrucción "DIM".

También pueden usarse tablas para variables alfanuméricas.

El espacio que en la memoria OCUPAN las tablas definidas mediante la clave de instrucción "DIM" puede ser LIBRADO, desocupado y hecho disponible, durante la ejecución del programa, usando la clave de instrucción "ERASE" (que corresponde a 'borrar'). Aquí hay un ejemplo:

```
10 DIM A(20)
20 ERASE A
30 DIM A(25)
```

Si suprimieras la línea 20 del programa en este ejemplo, MSX-BASIC al examinar la línea 30, produciría un mensaje de error dado que la tabla A ya tiene espacio ocupado para 21 variables. Al introducir la instrucción de la línea 20, hacemos que el espacio ocupado por la primera tabla quede libre, y luego ya podemos en la 30 hacer que se ocupe otro nuevo espacio en memoria correspondiente a 26 variables numéricas.

Decisiones

Capítulo 2.7 El 'trayecto' o recorrido seguido en la ejecución de un programa, puede variarse condicionalmente usando la clave de instrucción "IF", como en el siguiente ejemplo:

```
10 INPUT A,B
20 IF A=B THEN PRINT "A=B" ELSE PRINT "A<>B"
```

En este ejemplo, se comparan en la línea 20 los contenidos de las variables "A" y "B" Con la instrucción "IF", el MSX-BASIC es obligado a examinar SI la condición mencionada es cierta. Si sí lo es, se ejecuta la instrucción que va detrás de la palabra clave "THEN". Si no lo es, se ejecuta la instrucción identificada por la palabra clave "ELSE".

La palabra clave "ELSE" y las instrucciones subsiguientes en esa línea de programa son opcionales. Lo que significa que no tenemos que usarlas siempre. Si la línea no contiene la palabra clave "ELSE", el programa proseguirá con la siguiente línea de programa que corresponde, cuando la condición no sea cierta.

Si la palabra clave "THEN" viene seguida de la palabra clave "GOTO", tampoco necesita incluirse la palabra clave "THEN". Una alternativa es suprimir la palabra clave "GOTO", como se ilustra en el siguiente ejemplo:

```
10 INPUT A,B
20 IF A=B THEN 50 ELSE GOTO 40
30 END
40 PRINT "A<>B":GOTO 30
50 PRINT "A=B":GOTO 30
```

Dentro de la condición cuya certeza ha de ser examinada en la instrucción condicional "IF"..."THEN"..."ELSE"..., puedan incluirse los siguientes **signos de relación** o de comparación:

<	Menor que
>	Mayor que
=	Igual
<>	Distinto de
>=	Mayor o igual que
<=	Menor o igual que

Si en la condición mencionada está incluso un cálculo matemático, dicho cálculo se efectúa antes de examinar la certeza o falsedad de la condición. Ejemplo:

```
10 INPUT A,B
20 IF A MOD B=0 THEN PRINT "REST=0"
```

La instrucción que sigue a la clave "THEN" sólo se ejecuta si el resto de dividir "A" por "B" es igual a 0.

Usando la palabra clave "NOT", se invierte la certeza o falsedad de la condición examinada, tal y como se ilustra en el siguiente ejemplo:

```
10 INPUT A,B
20 IF NOT A=B THEN PRINT "A<>B"
30 IF A<>B THEN PRINT "A<>B"
```

Las condiciones de las líneas 20 y 30 del programa producen el mismo resultado.

Combinación de condiciones

Es posible incluir más de una condición, detrás de la palabra clave "IF", para hacer una combinación de condiciones o condición compleja, usando las palabras clave siguientes:

1. **Palabra conectiva: "AND" (YLACION o CONJUNCION)**

condición 1	**condición 2**	**resultado**
Cierto	Cierto	Cierto
Cierto	No Cierto	No cierto
No cierto	Cierto	No cierto
No cierto	No cierto	No cierto

2. **Palabra conectiva: "OR" (OLIACION o DISYUNCION)**

condición 1	**condición 2**	**resultado**
Cierto	Cierto	Cierto
Cierto	No Cierto	Cierto
No cierto	Cierto	Cierto
No cierto	No cierto	No cierto

3. **Palabra conectiva: "XOR" (OLEACION o DISCREPANCIA)**

condición 1	**condición 2**	**resultado**
Cierto	Cierto	No cierto
Cierto	No Cierto	Cierto
No cierto	Cierto	Cierto
No cierto	No cierto	No cierto

4. **Palabra conectiva: "EQV" (NOLEACION o COINCIDENCIA)**

condición 1	**condición 2**	**resultado**
Cierto	Cierto	Cierto
Cierto	No Cierto	No cierto
No cierto	Cierto	No cierto
No cierto	No cierto	Cierto

5. **Palabra conectiva: "IMP" (IMPLICACION)**

condición 1	**condición 2**	**resultado**
Cierto	Cierto	Cierto
Cierto	No Cierto	No cierto
No cierto	Cierto	Cierto
No cierto	No cierto	Cierto

Ejemplos son:

```
10 IF 3>1 AND 3>2 THEN PRINT "CORRECTO"
20 IF 3>1 OR 6<7 THEN PRINT "CORRECTO"
30 IF 3>1 XOR 6<1 THEN PRINT "CORRECTO"
40 IF 4>3 EQV 6<7 THEN PRINT "CORRECTO"
50 IF 4<3 IMP 3>2 THEN PRINT "CORRECTO"
```

En todas las líneas anteriores del programa se ejecutarán las instrucciones que siguen a la clave "THEN".

Comparando datos alfanuméricos

No sólo se puede usar en una condición de información numérica, sino también de información alfanumérica. La comparación se efectúa teniendo en cuenta los códigos de caracteres según el repertorio del 'home computer'. Encontrarás un resumen completo de todos los códigos de caracteres en el Apéndice E de este manual.
Ejemplo:

```
10 IF "A"<"B" THEN PRINT "A ES MENOR O ANTERIOR A B"
20 IF A$<B$ THEN PRINT "EL CONTENIDO DE A$ ES MENOR
   O ANTERIOR AL CONTENIDO DE B$"
30 IF A$="JUAN" THEN PRINT "EL CONTENIDO DE A$ ES JUAN"
40 IF "JOSE"<"JUAN" THEN PRINT "CORRECTO"
```

No puede usarse dentro de una condición sencilla, una mezcla de información numérica y alfanumérica.

Bucles en programas

Teclea el comando "NEW" y luego las siguientes instrucciones:

```
10 CT=1
20 PRINT "MSX-BASIC"
30 CT=CT+1
40 IF CT<11 GOTO 20
RUN
```

Cuando pases este programa, verás que el texto literal "MSX-BASIC" aparece 10 veces en tu pantalla. Pero hay otra forma más fácil de conseguir el mismo efecto:

```
NEW
10 FOR CT=1 TO 10
20 PRINT "MSX-BASIC"
30 NEXT CT
RUN
```

Cuando pasas este programa consigues exactamente el mismo resultado que en el ejemplo anterior.
En la línea 10 del programa, la variable numérica "CT" está dotada en la primera 'ronda' del bucle con el valor 1, luego, se expone el texto "MSC-BASIC" de acuerdo con la línea 20 de programa. En la línea 30 de programa, el valor de la variable "CT" se incrementa en 1; y si dicho valor es mayor que 10, se ejecutará la línea 40 de programa, pero mientras que sea menor o igual que 10, se repetirá la línea 30 del programa que es la que forma el llamado **bucle** cuyo inicio se señala en la línea 10 y cuyo final se señala en la línea 30.

En nuestro ejemplo, con la palabra clave "NEXT, la variable "CT" sólo se incrementa en una unidad. Ese valor de incremento puede cambiarse usando la cláusula opcional "STEP" dentro de la instrucción "FOR".

Los valores constantes que señalan el inicio y el término de la gama de valores de la instrucción "FOR", pueden sustituirse también por nombres de variables, como en nuestro ejemplo siguiente:

```
NEW
10 A=4
20 FOR CT=8 TO A STEP -2
30 PRINT "MSX-BASIC"
40 NEXT CT
RUN
```

En este ejemplo, el valor de nuestra variable "CT" no se incrementa al llegar a la instrucción "NEXT", sino que realmente se reduce en dos unidades, debido a la opinión "STEP". Eso significa que el valor inicial de nuestra variable "CT" tiene que ser mayor que el valor final. Si pasas el programa anterior, verás que el literal "MSX-BASIC" se expone tres veces en pantalla.

Ordenando (Sorting)

El siguiente ejemplo es un programa para ordenar una colección de números dados por el usuario, colocándolos en secuencia ascendente. El usuario tiene que hacer saber al ordenador con cuántos números ha de trabajar y luego imponerle los números pertinentes. Los números se conservan en una tabla de variables y se ordenan de acuerdo con el llamado método binario.

```
NEW
10 DIM G(100)
20 INPUT "DIME CUANTOS NUMEROS (ENTRE 2 Y 100)";A
30 IF (A<2)OR(A>100) GOTO 20
40 FOR I=1 TO A
50 PRINT "NUMERO";I;
60 INPUT G(I)
70 NEXT I
75 REM CUANDO I ES MAS PEQUEÑO QUE EL CONTENIDO DE A,
   LA EJECUCION PROSEGUIRA CON EL NUMERO DE LINEA 50
80 PRINT "ESTOY ORDENANDO LOS NUMEROS QUE ME HAS DADO"
90 FOR I=2 TO A
100 X1=1:X2=I
110 X3=X1+INT((X2-X1)/2)
120 IF G(I)<G(X3) THEN 150
130 IF G(I)>G(X3) THEN 170
140 X2=X3
150 IF X2=X3 THEN 190
160 X2=X3:GOTO 110
170 IF X1=X3 THEN 190
180 X1=X3:GOTO 110
190 H=G(I)
200 FOR J=I TO X2+1 STEP -1
210 G(J)=G(J-1)
220 NEXT J
230 G(X2)=H
240 NEXT I
245 REM CUANDO I ES MAS PEQUEÑO QUE EL CONTENIDO DE A,
    LA EJECUCION PROSEGUIRA CON LA LINEA NUMERO 100
250 PRINT "ESTOS SON LOS NUMEROS ORDENADOS EN SECUENCIA"
260 FOR I=1 TO A
270 PRINT G(I);
280 NEXT I
```

Si ejecutas este programa, verás que los números están puestos en secuencia, ordenados. Si tu programa no funciona apropiadamente, comprueba hasta descubrir dónde has tecleado algo incorrecto.

Programas con menú

Los programas capaces de efectuar varias tareas, comienzan a menudo con un "recetario", o **menú de elección**. El menú muestra un índice de las secciones del programa y cuál es la sección particular del programa que se ejecuta cuando se elige un número determinado. Aquí hay un ejemplo de tal clase de programas:

```
NEW
10 PRINT "MENU A ELEGIR"
20 PRINT "1=LINEA 100 DE PROGRAMA"
30 PRINT "2=LINEA 200 DE PROGRAMA"
40 PRINT "3=CONCLUSION DE PROGRAMA"
50 INPUT "DIME TU ELECCION";A
60 IF (A<1)OR(A>3) GOTO 50
70 ON A GOTO 100,200,300
100 PRINT "AQUI EMPEZARIA LA SUBRUTINA 100":GOTO 10
200 PRINT "AQUI EMPEZARIA LA SUBRUTINA 200":GOTO 10
300 END
RUN
```

La línea más interesante de este programa es la 70, que hace saltar SEGUN el contenido de la variable "A". Si el valor de "A" es igual a 1, el programa continúa su recorrido con la línea 100. Si es igual a 2, el programa continuará con la línea 200, y cuando es 3, el programa saltará a la línea 300 y terminará.

Tratamiento de errores

MSX-BASIC hace posible detectar y tratar los errores cometidos durante la ejecución de un programa. Aquí hay un ejemplo:

```
NEW
10 ON ERROR GOTO 50
20 INPUT "NUMERO";A%
30 END
50 IF ERR=6 THEN PRINT "DEBIERA SER UN NUMERO ENTRE
   -32768 Y +32767":RESUME 0
60 ON ERROR GOTO 0
RUN
```

En este ejemplo, la instrucción de la línea 10 hace que el programa vaya a la línea 50, siempre que se detecte un error en la ejecución. Si el usuario teclea en la línea 20 una cantidad menor de 32768 o mayor de 32767, se producirá un error por haberse REBASADO la gama permitida, dado que la variable con nombre "A%" designa una variable entera. Este error de rebase, tiene **código de error** 6 (véase el Apéndice A de este manual). Al pasar a la línea 50, como consecuencia del error, se examina a ver si realmente ha sido el código 6 de error al que ha suscitado la **rutina** de tratamiento de errores. Si así ha sido, se presenta en pantalla el texto mencionado en la línea 50, y después de eso, debido a la instrucción "RESUME", se REANUDA la ejecución del programa comenzando a partir de la primera línea. Debido a la línea 60, MSX-BASIC interrumpirá la ejecución del programa si el error producido corresponde a un código distinto del que se examina en la línea 50, por ejemplo, tecleando una letra.

MSX-BASIC también permite que el propio usuario defina sus códigos de error. Todo lo que hay que tener en cuenta, es que los códigos de error a usar deben ser mayores que los ya empleados por MSX-BASIC. Aquí hay un ejemplo:

```
NEW
10 ON ERROR GOTO 100
20 INPUT "CONCLUSION DE PROGRAMA";A$
30 IF A$="SI" THEN ERROR 250
40 PRINT "EJECUCION DEL PROGRAMA":GOTO 20
50 END
100 IF ERR=250 THEN INPUT "ESTAS SEGURO";A$:IF A$="SI"
    THEN RESUME 50 ELSE RESUME 20
110 ON ERROR GOTO 0
RUN
```

En la línea 30 de programa, se define el código de error 250 para las ocasiones en que la variable A$ tiene como valor el literal "SI". En ese caso, al ejecutarse la línea 100, como consecuencia de cualquier error producido en la ejecución —tal y como se define en la línea 10 para CUANDO HAYA error— se examinará para ver si el error corresponde al código 250. Si sí corresponde, MSX-BASIC pedirá al usuario confirmación, y si la respuesta de éste es positiva, el programa reanudará en la línea 50 debido a la instrucción "RESUME 50"; pero si la respuesta del usuario es negativa, el programa reanudará la ejecución en la línea 20, debido a la instrucción "RESUME 20".

Subrutinas

Capítulo 2.8

Supongamos que tenemos en memoria el siguiente programa:

```
10 INPUT A:INPUT B
20 C=A*100/B:PRINT A;
30 PRINT "ES";C;"POR CIENTO DE";
40 PRINT B
50 C=B*100/A:PRINT B;
60 PRINT "ES";C;"POR CIENTO DE";
70 PRINT A
80 END
```

Sin ninguna duda, habrás observado que la línea 60 de programa es idéntica a la línea 30. Puedes además, usar una línea especial de programa y mencionar al ordenador esa línea cuando llegue a la línea 30 y a la línea 60, si usas la clave de instrucción "GOSUB", que hace que VAYA a esa línea especial y VENGA cuando haya ejecutado.

Nuestro programa ejemplo, sería entonces el siguiente:

```
10 INPUT A:INPUT B
20 C=A*100/B:PRINT A;
30 GOSUB 90
40 PRINT B
50 C=B*100/A:PRINT B;
60 GOSUB 90
70 PRINT A
80 END
90 PRINT "ES";C;"POR CIENTO DE";
100 RETURN
```

Cuando llega a la línea 30, MSX-BASIC proseguirá su camino desviándose a la línea 90 y ejecutando lo que allí se le manda. La instrucción "RETURN" que viene detrás, hace que VUELVA al punto donde se 'desvió' y ejecute la línea siguiente, en este caso la 40. Lo mismo sucede cuando llega a la línea 60, pero esta vez la VUELTA hace que pase a ejecutar la línea a continuación del desvío, la línea 70.

La sección de programa formada por las líneas 90 y 100 se denomina **subrutina**, y siempre debe estar terminada por una instrucción de VUELTA (RETURN). Cuando MSX-BASIC llega a la línea 90 como resultado de haber sido **citada** la subrutina mediante una instrucción "GOSUB", se anota el punto donde se le cita para poder volver a él cuando llegue a la instrucción "RETURN" que termina la subrutina. La línea 80 de programa es de importancia particular cuando se utilizan subrutinas. Si la olvidas, y la línea 90 viniera inmediatamente detrás de la 70, MSX-BASIC daría el mensaje de error "RETURN without GOSUB" al encontrarse la línea 100 que le manda volver sin que ninguna le haya mandado que VAYA y VENGA al acabar.

Subrutinas suscitadas en otras ocasiones

Además de poder citar directamente una subrutina mediante la instrucción "GOSUB", se puede **suscitar** la ejecución de cualquier subrutina **según** determinadas informaciones tales como:

1. El contenido de una variable.
2. La hora o tiempo transcurrido.
3. Pulsación de una o más de las teclas funcionales.
4. La pulsación simultánea de CTRL y STOP.
5. Colisiones entre dos 'sprites'.
6. El botón accionador de un joystick de juegos.

Estas dos últimas ocasiones, se comentan con más detalle en los capítulos 2.12 y 2.14 de este manual.

Una subrutina puede ser citada según el contenido de una variable usando la clave de instrucción "ON". Aquí hay un ejemplo:

```
10 INPUT A
20 IF (A<0)OR(>255) GOTO 10
30 ON A GOSUB 100,200,300
40 END
100 PRINT "SUBRUTINA 100":RETURN
200 PRINT "SUBRUTINA 200":RETURN
300 PRINT "SUBRUTINA 300":RETURN
```

En este ejemplo, al llegar a la línea 30, se ejecutará una determinada subrutina SEGUN el valor de la variable "A": si "A"=1 se ejecutará la subrutina que comienza en la línea 100; si "A"=2, la que empieza en la 200, y si "A"=3 la que comienza en la 300. El contenido de la variable "A" en la línea 30, no puede ser negativo ni mayor de 255, porque MSX-BASIC produciría un mensaje de error. Obviamente, en lugar de la variable "A" se puede mencionar en la instrucción cualquier fórmula o expresión matemática, tal como:

```
30 ON A+1 GOSUB 100,200,300
```

En este caso, cuando el contenido de la variable "A" es 1, se ejecutará la línea 200, porque "A"+1 tendría el valor 2. Con respecto a los cálculos, se aplica la misma regla anterior, es decir: el resultado no puede ser negativo ni mayor de 255. Cuando el valor de la variable "A" o el resultado del cálculo es mayor que la cantidad de números de línea mencionados después de la clave "GOSUB", lo que se ejecuta es la siguiente instrucción. Si, por ejemplo, nuestra variable fuera "A"=3, en este caso se ejecutaría la línea 40, porque "A"+1=4 y solamente hemos mencionado tres subrutinas.

Si la ejecución de una subrutina se suscita por la indicación del **tiempo** transcurrido, teclas funcionales o pulsación de CTRL más STOP, estos desvíos provocados hacia las subrutinas tienen que estar preparados de antemano. Observa el siguiente ejemplo:

```
10 ON STOP GOSUB 100
20 ON INTERVAL=250 GOSUB 200
30 ON KEY GOSUB 300
40 STOP ON:INTERVAL ON:KEY(1) ON
50 GOTO 50
100 PRINT "FIN":END
200 PRINT "HAN PASADO OTROS 5 SEGUNDOS"
210 RETURN
300 PRINT "HA SIDO PULSADA LA TECLA F1"
310 RETURN
```

En el ejemplo anterior, las líneas 10, 20 y 30 del programa indican a MSX-BASIC las subrutinas que tendrá que ejecutar CUANDO se pare la ejecución del programa porque se han pulsado simultáneamente las teclas CTRL y STOP, tal y como se prepara en la línea 10. Cuando haya transcurrido un intervalo de 250 x 1/50 segundos, tal y como se prepara en la línea 20. Cuando se haya pulsado cualquiera de las teclas funcionales.

Observa, sin embargo, que con estas instrucciones únicamente se "prepara un cepo" para atrapar ciertos eventos y se designa la subrutina a ejecutar cuando se produzcan dichos eventos; pero no se dispararán dichos cepos, a no ser que sean **armados** mediante instrucciones tales como las que figuran en la línea 40 de programa.

La línea 50 de programa parece ser absolutamente innecesaria, dado que significa que MSX-BASIC debe continuar ejecutando un salto a la línea 50 de manera indefinida. Pero mientras lo hace, MSX-BASIC comprobará si una de las teclas ha sido pulsada, o si ha transcurrido el intervalo exigido, momento en que se disparará el cepo y se suscitará la ejecución de la rutina asociada a cada uno.

Presta atención especial al hecho de que la subrutina que comienza en la línea 100 no está terminada como es habitual con la clave "RETURN", sino con la clave de instrucción "END". Si así no lo hicieramos, sería imposible detener la ejecución del programa sin apagar el ordenador.

¿Qué día es?

En nuestro siguiente ejemplo, MSX-BASIC determinará el día de la semana que corresponde a una fecha específica que impongamos. Primero, vacía la memoria con el comando "NEW" y luego teclea el siguiente programa:

```
10 DATA DOMINGO,LUNES,MARTES,MIERCOLES,JUEVES,
   VIERNES,SABADO
20 PRINT "DIME LA FECHA"
30 INPUT "DIA (1-31)";DD
40 IF (DD<1) OR (DD>31) GOTO 30
50 INPUT "MES (1-12)";MM
60 IF (MM<1) OR (MM>12) GOTO 50
70 INPUT "AÑO (1-3000)";AA
80 IF (AA<1) OR (AA>3000) GOTO 70
90 IF MM>2 GOTO 110
100 AA=AA-1::MM=MM+12
110 H1=INT(AA/100):H2=AA-H1*100
120 H3=INT(2.6001*(MM-2)-.2)+DD+H2+INT(H2/4)+
    INT(H1/4)-2*H1
130 H3=H3-INT(H3/7)*7+1
```

```
135 REM LAS LINEAS 100 A 130 ESTAN ESPECIALMENTE
    DIRIGIDAS A ESTA CLASE DE PROGRAMA
140 RESTORE
150 FOR I=1 TO H3:READ H$:NEXT I
160 PRINT "ESE DIA CAE EN ";H$
170 END
```

Ejecuta este programa y teclea la fecha en la forma que te pregunta. Si el programa no funciona apropiadamente, comprueba cuidadosamente si has cometido un error al teclear la serie de instrucciones.

Funciones

Capítulo 2.9 Además de claves de comandos y de instrucciones, MSX-BASIC también permite un gran número de **funciones**, designadas por palabras clave específicas. Estas funciones entregan el resultado del cálculo efectuado sobre una constante o sobre una variable que actúa y recibe el nombre de **argumento**. Sólo pueden usarse cuando la frase correspondiente está precedida por una clave de instrucción o de comando. Aquí hay un ejemplo:

```
NEW
10 PRINT SIN(5)
RUN
-.9589242746631
```

o bien, otro ejemplo:

```
NEW
10 X=5
20 PRINT SIN(X)
RUN
-.9589242746631
```

Encontrarás en el manual de referencia un resumen completo de todas las funciones disponibles. Pueden resumirse en diferentes clases:

1. **Funciones matemáticas**
   "ABS", "ATN", "COS", "EXP", "INT", "LOG", "SGN", "SIN", "SQR", "TAN".
2. **Funciones alfanuméricas**
   "ASC", "LEFT$", "LEN", "MID$", "RIGHT$", "STR$", "STRING$", "VAL", "BIN$", "HEX$", "OCT$", "INSTR$".
3. **Funciones de conversión**
   "CDBL", "CINT", "CSNG", "FIX".
4. **Funciones de entrada de datos**
   "INKEY$", "INPUT$".
5. **Funciones de salida de datos**
   "CHR$", "POS", "LPOS", "SPC", "TAB".
6. **Funciones varias**
   "CSRLIN", "ERL", "ERR", "FRE", "PEEK", "RND", "USR", "VARPTR", "POINT", "VPEEK", "STICK", "STRIG", "PDL", "PAD", "PLAY", "EOF".

El usuario puede además definir sus propias funciones, usando la clave de instrucción "DEF FN", tal como se explica en el capítulo 2.4.

Los números aleatorios

La clave de función "RND" se usa a menudo para hacer que MSX-BASIC determine un número **aleatorio**. Ese número estará entre 0 y 1, y si el usuario desea, por ejemplo, un número entre 1 y 6 (resultado del lanzamiento de un dado, por ejemplo), simplemente tiene que multiplicar por 6 el número obtenido y añadir 1 al resultado intermedio. Aquí hay un ejemplo:

```
NEW
10 A=INT(RND(1)*6+1)
20 PRINT A
RUN
```

En el ejemplo anterior, hemos usado dos funciones combinadas una como argumento de otra en la línea 10. La clave de función "INT" hace que se obtenga como resultado un número que sólo tiene la parte entera del número seleccionado aleatoriamente.

Números de lotería

Nuestro siguiente ejemplo es un programa que elige 6 números de lotería.

```
NEW
10 X=0
20 FOR I=1 TO 6
30 A=INT(RND(1)*49+1)
50 IF A=X(1) GOTO 30
60 IF A=X(2) GOTO 30
70 IF A=X(3) GOTO 30
80 IF A=X(4) GOTO 30
90 IF A=X(5) GOTO 30
110 X(I)=A
120 NEXT I
130 PRINT "LOS NUMEROS DE LOTERIA SON";
140 FOR I=1 TO 6:PRINT X(I);:NEXT I:PRINT
160 INPUT "DOY OTRA RONDA (S/N)";A$
170 IF A$="S" GOTO 10
180 END
```

Ejecuta este programa y verás los números de lotería que MSX-BASIC genera para ti. Si el programa no funciona apropiadamente, comprueba cuidadosamente si has tecleado las instrucciones en la forma adecuada.

Espacio de memoria

La clave de función "FRE" te permite saber de cuánto espacio de memoria puedes disponer todavía. Ensaya el comando:

```
PRINT FRE(0)
 nnnnn
```

Y el número representado por nnnnn te indicará en octetos la cantidad de memoria LIBRE.

Si lo que deseas saber es cuánto espacio de memoria queda libre para ser usado en variables alfanuméricas usa el comando:

```
PRINT FRE("")
 xxx
```

Y el número obtenido representará en octetos el espacio libre para variables alfanuméricas.

MSX-BASIC y la pantalla

Capítulo 2.10

Con MSX-BASIC podemos usar la pantalla de cuatro **modos** diferentes:

1. Modo 1 de texto.
2. Modo 2 de texto.
3. Modo 1 gráfico.
4. Modo 2 gráfico.

Modo 1 de texto

En este modo se pueden situar 40 caracteres como máximo en una línea de la pantalla (40 columnas), y pueden aparecer 24 líneas en pantalla. Cada una de las 960 posiciones diferentes en la pantalla, queda definida dando la **coordenada "x"** (indicando la columna) y la **coordenada "y"** (indicando la fila o renglón). El valor mínimo de la coordenada "x" es 0 (cero) y corresponde a la columna más a la izquierda; mientras que el valor máximo es 39 y corresponde a la columna extremo-derecha. Los valores de la coordenada "y" van desde 0, correspondiente a la línea superior, hasta 23 que corresponde a la línea inferior. Con la clave de instrucción "WIDTH", puedes decirle al ordenador cuántos caracteres (columnas) deseas que haya en cada línea.

Con la clave de instrucción "LOCATE", puedes situar el **cursor de texto** en cualquier posición permitida de la pantalla.

El modo 1 de texto puede conseguirse con la clave de instrucción "SCREEN 0". Aquí hay un ejemplo:

```
NEW
10 SCREEN 0
20 WIDTH 40
30 LOCATE 15,10:PRINT "HOLA"
RUN
```

En este ejemplo, el modo 1 de texto se elige en la línea 10. La línea 20 abre la posibilidad de situar 40 caracteres en cada línea. En la línea 30 el cursor es situado en la posición 15 como coordenada "x" (16ª columna) de la línea número 10 (11ª línea), y a partir de esta posición se expondrá el mensaje.

Cuando MSX-BASIC está a nivel de comando, o tiene que ejecutar una instrucción de imposición de datos (clave "INPUT"), se coloca siempre y automáticamente en el modo 1 de texto.

Modo 2 de texto

Este modo de texto permite un máximo de 32 caracteres por cada línea de pantalla, que siguen siendo 24 posibles. Cada posición de la pantalla también se define dando la coordenada "x" (la columna) y la coordenada "y" (la fila o renglón). La coordenada "x" es un número entre 0 para la columna extremo-izquierda y 31 para la columna extremo-derecha. La coordenada "y" va desde 0 para la línea superior a 23 para la línea inferior.

El número máximo de caracteres por línea, la **anchura,** puede determinarse usando la clave de instrucción "WIDTH".

Usando la clave de instrucción "LOCATE", puede **ubicarse** el cursor en cualquiera de las posiciones de la pantalla que se desee.

Este modo 2 de texto se elige usando la clave de instrucción "SCREEN 1". Aquí hay un ejemplo:

```
NEW
10 SCREEN 1
20 WIDTH 32
30 LOCATE 15,10:PRINT "HOLA"
RUN
```

Este ejemplo conduce exactamente al mismo resultado que el ejemplo anterior, presentado al hablar del modo 1 de texto en pantalla.

En la línea 10 de programa, sin embargo, hemos elegido ahora el modo 2 de texto. En la línea 20, la posibilidad de usar anchura de 32 caracteres por línea. Si tienes dificultad en leer en la pantalla del monitor los caracteres usando el modo 1 de texto, pasa al modo 2 de texto que es más legible.

Observa por favor, que hay un área rodeando al texto en color ciano (el 'limbo' o 'marco') en la parte superior e inferior de la pantalla en el modo 2 de texto. Puedes cambiar el color de estas áreas que bordean el texto, usando la clave de instrucción "COLOR".

Para volver al modo 1 de texto, teclea:

```
SCREEN 0:WIDTH 40
```

Modo 1 gráfico

En este modo gráfico, la pantalla está formada por una **red de malla,** con 256 puntos horizontales y 192 puntos verticales. La posición de cada punto viene como siempre determinada por el valor de la coordenada "x" (la columna) y el valor de la coordenada "y" (la fila o renglón). Por tanto, la coordenada "x" puede ser cualquier número entre 0 para la columna extremo-izquierda y 255 para la columna extremo-derecha. La coordenada "y" puede ir desde 0 para la línea superior, hasta 191 para la línea inferior.

En este modo, solamente pueden usarse instrucciones gráficas para componer una imagen en la pantalla. Se elige mediante la clave de instrucción "SCREEN 2", y las instrucciones gráficas se detallan en el capítulo 2.11.

```
NEW
10 SCREEN 2
20 CIRCLE (185,115),10
30 GOTO 30
RUN
```

En este ejemplo, el 'home computer' dibujará un círculo en la pantalla del monitor, usando como centro el punto situado en la columna 186 (coordenada x = 185) y en la fila 116 (coordenada y = 115), con un radio de 10 **motas,** o puntos de imagen.

La línea 30 es aquí de particular importancia, ya que al usarla se ejecuta repetidamente la ida a dicha línea. Si no se hubiera incluido la línea 30, MSX-BASIC hubiera vuelto a nivel de comando después de ejecutar las líneas 10 y 20, pasando automáticamente al modo 1 de texto. Considerando la velocidad a que opera MSX-BASIC, no hubieras tenido tiempo de ver el círculo que dibuja en pantalla.

La única manera de que pare la ejecución del programa es pulsando simultáneamente las teclas CTRL y STOP.

Modo 2 gráfico

En este modo, podemos considerar que la pantalla está dividida exactamente de la misma manera que en el modo 1 gráfico. La diferencia radica en que cada imagen se compone de pequeños cuadraditos que miden siempre 4 x 4 **motas** o puntos de imagen. Cuando se elige el modo 2 gráfico, con la clave de instrucción "SCREEN 3", sólo se pueden usar instrucciones y comandos gráficos. Encontrarás detalles completos sobre las diversas claves de instrucciones gráficas en el capítulo 2.11.

Observemos un ejemplo:

```
NEW
10 SCREEN 3
20 LINE (125,10)-(200,85)
30 GOTO 30
RUN
```

En este ejercicio se dibujará una línea recta desde el punto 125 (coordenada "x"), 10 (coordenada "y") hasta el punto 200 (coordenada "x"), 85 (coordenada "y"). Y dicha línea estará formada por pequeños cuadraditos que ocupan 4 x 4 puntos de imagen. Puedes hacer que pare el programa pulsando simultáneamente las teclas CTRL y STOP.

Veamos ahora lo que sucede cuando cambiamos la línea 10 para que sea:

```
10 SCREEN 2
```

Al ejecutar ahora el programa, veremos la diferencia entre los tamaños de puntos correspondientes a los modos 1 y 2 gráficos.

Limpiando la pantalla

Puedes **limpiar** la pantalla, quitando la imagen presente en ella, usando la clave de instrucción "CLS" que puede usarse en cualquiera de los modos de pantalla.

Instrucciones para color y gráficos

Capítulo 2.11. MSX-BASIC permite el uso de 16 colores. Estos colores están numerados del 0 al 15. Pueden usarse como colores **de frente** y **de fondo,** y también para dar color al área de borde **(limbo),** en el modo 2 de texto y en los modos 1 y 2 gráficos.

La lista completa de los diversos colores se da en el Apéndice C.

El color marcado como número 0 **(transparente)** tiene interés particular, porque cuando se usa este número de color, sólo aparecerá el color predominante del fondo, en aquellas áreas cubiertas por un color del frente transparente. Eso significa que el color número 0 sólo se usa cuando se necesita como color del frente; y cuando se elige el color transparente para el frente y el fondo, la mota correspondiente aparecerá en negro.

Los colores se eligen usando la clave de instrucción "COLOR". Vacía la memoria del ordenador con el comando "NEW" y teclea el siguiente ejemplo:

```
10 SCREEN 1
20 COLOR 1,3,6
30 PRINT "HOLA"
RUN
```

En este ejemplo, las letras aparecen en negro (color del frente con número 1) proyectadas sobre un fondo verde claro (color del fondo número 3), y el área que bordea la imagen aparece en rojo oscuro (color número 6).

Cuando hayas estudiado ese ejemplo, teclea la línea de comandos:

```
SCREEN 0:COLOR 15,4,7
```

Al comenzar a trabajar con MSX-BASIC, los colores prescritos para el modo 1 de texto son blanco para el frente (número 15) y azul oscuro para el fondo (color número 4). En los otros modos de pantalla, también verás que las áreas que bordean la imagen aparecen en color ciano (número 7) como color prescrito o estándar.

Dibujando una línea

Con la clave de instrucción "LINE" puedes dibujar una línea recta entre dos puntos cualesquiera de la pantalla. Esos puntos deberán ser señalados por sus coordenadas "x" e "y" sea en forma **absoluta** o en forma **relativa**. El valor relativo de las coordenadas se indica precediendo el salto, o **incremento,** con la palabra clave "STEP". Un punto por ejemplo, en la columna 11, fila 16, tiene como coordenadas absolutas los valores (10,15). Supongamos que las coordenadas absolutas del último punto pintado en pantalla son (3,6), entonces, el punto que estamos tratando de coordenadas absolutas (10,15) podría señalarse en coordenadas relativas mediante STEP (7,9).

```
NEW
10 SCREEN 2
20 LINE (125,10)-STEP(75,75),1
30 GOTO 30
RUN
```

En este ejemplo, las coordenadas del segundo de los puntos de la línea recta se dan en relación, relativas al primero de los puntos de la recta. El **parámetro** final "1" en la línea 20 de programa, determina el color de la línea recta pintada. En este caso, negro (número de color 1). Eso significa que la línea 20 de programa podríamos también escribirla dando las coordenadas absolutas del segundo de los puntos, como sigue:

```
20 LINE (125,10)-(200,85),1
```

Si añadimos la letra clave "B" después del número de color en la instrucción que pinta líneas, MSX-BASIC producirá un **rectángulo** usando esa línea como diagonal del mismo.

La línea por supuesto, no aparece en pantalla. Detén el programa y efectúa el siguiente cambio:

```
20 LINE (125,10)-STEP(75,75),1,B
```

y ejecuta de nuevo el programa.

Si se incluyen las letras clave "BF" después del número de color de una instrucción que pinta líneas, MSX-BASIC dibujará un rectángulo en la pantalla y luego **rellenará** el recinto del mismo de acuerdo con el color mencionado en la instrucción de tirar líneas.

Detén el programa y cambia la línea 20 para que sea:

```
20 LINE (125,10)-STEP(75,75),1,BF
```

Ahora ejecuta el programa de nuevo y observa el efecto. Para que pare, pulsa simultáneamente las teclas CTRL y STOP.

Pintando un arco de circunferencia

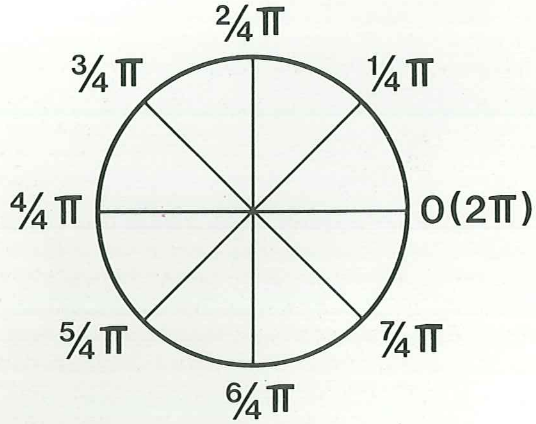
Con la clave de instrucción "CIRCLE" puedes dibujar arcos de circunferencias, circunferencias completas o círculos, tomando como centro un punto determinado por sus coordenadas "x" e "y". Como siempre, esas coordenadas pueden venir dadas en forma **absoluta,** o en forma **relativa.** El radio del arco viene dado en **motas,** o 'puntos de imagen', según el modo gráfico en que se esté trabajando. Aquí hay un ejemplo:

```
NEW
10 SCREEN 2
20 CIRCLE (90,80),20,1
30 GOTO 30
RUN
```

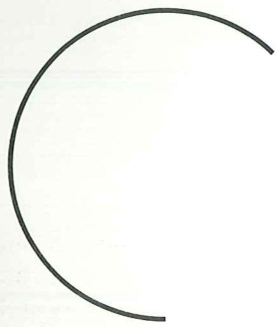
En este ejemplo, se pinta una circunferencia en negro (número de color 1) alrededor del centro que es el punto 90 (coordenada x), 80 (coordenada y), con un radio de 20 motas, o puntos de imagen.

Con la clave de instrucción "CIRCLE" también puede pedirse que pinte un arco de circunferencia dando el ángulo de principio de la curva y el ángulo de final de la curva. Ambos se determinan de acuerdo con el siguiente diagrama:

```
Π = 3.1415926535898
```



Supongamos que queremos dibujar el arco de circunferencia indicado en esta otra figura:



Detén primero la ejecución de nuestro programa anterior y luego cambia la línea 20 para que sea:

```
20 CIRCLE (90,80),20,1,0.7854,4.7124
```

Ahora ejecuta el nuevo programa. El punto del extremo inicial de la curva viene dado por 3.1416 x 1/4 = 0.7854 y el punto del extremo final de la curva por 3.1416 x 6/4 = 4.7124.

Los puntos de comienzo y terminación de una curva, pueden conectarse al centro del arco, formando así sectores, si expresamos los ángulos de comienzo y terminación como valores **negativos.** Detén el programa anterior (teclas CTRL+STOP) y cambia la línea 20 para que sea:

```
20 CIRCLE (90,80),20,1,-0.7854,-4.7124
```

Verás cómo se trazan los dos radios correspondientes a los puntos de comienzo y terminación; y obviamente, también es posible conectar sólo uno de los extremos de la curva al centro.

MSX-BASIC puede también pintar arcos de elipses, y elipses completas, con la clave de instrucción "CIRCLE". Para hacerlo, hay que introducir otro parámetro que indica la proporción entre los ejes horizontales y verticales de la elipse. Detén la ejecución del programa anterior y cambia la línea 20 para que sea:

```
20 CIRCLE (90,80),20,1,,,2
```

Asegúrate que has tecleado exactamente todas las comas antes de ejecutar el nuevo programa. Si no lo haces, MSX-BASIC al examinar la instrucción, considerará que el parámetro "2" que has incluido, corresponde al ángulo de comienzo de la curva y no a la proporción entre los diámetros horizontal y vertical. En este ejemplo, el eje vertical será 20 y el horizontal será 20 : 2 = 10, con lo que saldrá una elipse alargada verticalmente.

Rellenando de color

Con MSX-BASIC es posible rellenar de color un recinto limitado por una curva cerrada, usando la clave de instrucción "PAINT". Junto con esta clave ha de mencionarse la posición en coordenadas "x" e "y" de un punto interior del recinto, además del número de color correspondiente a la curva que lo limita. Las coordenadas pueden darse tanto en valor absoluto como en valor relativo, y a partir de ese punto, MSX-BASIC buscará un área que encerrando al punto, esté limitada por una curva pintada en el color mencionado en la instrucción, y rellenará dicho área con ese color. Ensaya el ejemplo siguiente:

```
NEW
10 SCREEN 2
20 CIRCLE (50,80),30,1
30 CIRCLE (90,80),30,1
40 PAINT (70,80),1
50 GOTO 50
RUN
```

Al ejecutar este programa, MSX-BASIC trazará dos circunferencias en negro que se solapan parcialmente. Con la instrucción de la línea 40 situamos el 'pincel' dentro del área de solape de ambas circunferencias y la rellenamos de color negro (número de color 1 en la línea 40). Ahora, detén la ejecución y cambia la línea 20 para que sea:

```
20 CIRCLE (50,80),30,6
```

Y ejecuta el nuevo programa. Ahora, con la instrucción 40 rellenamos el círculo trazado por la instrucción 30, dado que es el único recinto delimitado por líneas en negro.

Observa que el punto de intersección de dos líneas siempre adopta el color de la última instrucción gráfica ejecutada. En nuestro ejemplo, la circunferencia roja trazada por la instrucción 20, ya no forma una línea continua, sin interrupciones. Intentar rellenar ese recinto con un color en particular, tendrá consecuencias aparentemente extrañas. Detén primero la ejecución del programa y cambia la línea 40 para que sea

```
40 PAINT (70,80),6
```

Ahora ejecuta el nuevo programa y verás que MSX-BASIC rellena toda la pantalla de rojo, dado que no hay ningún área completamente cerrada por líneas rojas. El punto de intersección de ambas circunferencias forma un 'poro' por donde se cuela el color.

En el modo 2 gráfico, MSX-BASIC puede buscar como cierre del recinto el color que bordea la imagen, tal y como hacemos en el siguiente ejemplo:

```
NEW
10 SCREEN 3
20 LINE (125,10)-STEP(75,75),15,B
30 PAINT (150,15),6,15
40 GOTO 40
RUN
```

Ahora verás que el cuadrado formado por una línea blanca (línea 20 del programa) se colorea de rojo debido a la línea 30 de programa. Al trabajar en el modo 2 gráfico, la instrucción de rellenar recintos puede diferir del color con que se han pintado las curvas que encierran un área dada.

Pintando una malla

Con la clave de instrucción "PSET" se puede pintar de un cierto color una **malla** de la red o gradilla de la pantalla, un solo punto de imagen. La clave de instrucción "PRESET" puede usarse para hacer que una malla desaparezca o recobre el color de fondo que tenía. Puedes verlo en el siguiente ejemplo

```
NEW
10 SCREEN 2
20 CIRCLE (80,80),20,1
30 FOR I=I TO 1000
40 PSET (80,80),15
50 PRESET (80,80)
60 NEXT
RUN
```

En este ejemplo, el punto central de la circunferencia negra se resalta haciéndola que alternativamente la malla sea blanca y azul oscuro. Al estar trabajando en el modo gráfico 2, la malla ocupa 4 x 4 motas o puntos de imagen.

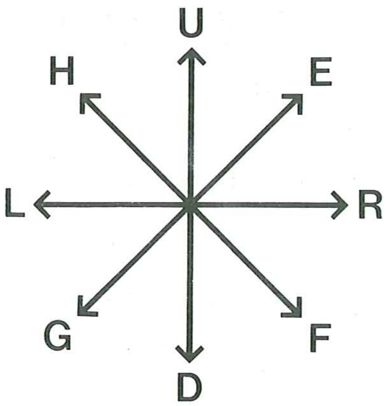
Trazando

Puedes formar figuras a **trazos** con MSX-BASIC mediante la clave de instrucción "DRAW". Es preciso determinar los trazos usando diversas letras clave, o subcomandos:

1. **Subcomandos para trazar rectas**

   A partir de la posición presente del cursor de gráficos, puedes trazar una recta según ocho direcciones diferentes:

   Un — Norte
   Dn — Sur
   Ln — Oeste
   Rn — Este
   En — Nordeste
   Fn — Sudeste
   Gn — Suroeste
   Hn — Noroeste



El parámetro "n" de estos subcomandos gráficos determina la longitud de la recta en términos de puntos de imagen.

```
NEW
10 SCREEN 2
20 PSET (80,80),15
30 DRAW "E50"
40 GOTO 40
RUN
```

Verás que MSX-BASIC traza una recta en diagonal, en dirección Nordeste. Observa que los subcomandos o parámetros de la instrucción "DRAW" han de expresarse como un valor alfanumérico lo que significa que debes encerrarlas entre comillas siempre, ya que se considera como constante alfanumérica.

Para trazar una línea desde la presente posición del cursor de gráficos a otra posición cualquiera de la pantalla, debes usar el subcomando "MXY" siendo X e Y las coordenadas del extremo de dicha recta. Estas coordenadas pueden venir dadas en valores absolutos o en valores relativos al último punto tratado. Si se usan valores relativos, deben estar precedidas por el signo mas (+) o el signo menos (—) según corresponda al incremento de posición efectuado. Los siguientes dos ejemplos producen el mismo efecto. Detén la ejecución del programa corriente (CTRL+STOP), y teclea:

```
NEW
10 SCREEN 2
20 PSET (80,80),15
30 DRAW "M90,75"
40 GOTO 40
RUN
```

Detén la ejecución y cambia la línea 30 para que sea:

```
30 DRAW "M+10,-5"
```

Y ahora observa que tienes el mismo efecto.

Todos los subcomandos gráficos anteriores pueden precederse por las letras clave N y B. Si usas el subcomando "N" el cursor de gráficos, el 'pincel', volverá a su posición primitiva después de ejecutar la instrucción de trazado. Teclea el siguiente ejemplo

```
NEW
10 SCREEN 2
20 PSET (80,80),15
30 DRAW "R10D10"
40 GOTO 40
RUN
```

En este ejemplo, MSX-BASIC trazará una recta hacia la derecha (R10) y luego desde el extremo de ella, una recta hacia abajo (D10). Detén la ejecución y cambia la línea 30 para que sea:

```
30 DRAW "NR10D10"
```

Ahora pasa el nuevo programa y verás que esta vez, primero traza una recta a la derecha (R10), y luego vuelve a colocar el pincel en el punto primitivo en que empezó dicha recta, antes de dibujar la segunda recta hacia abajo (D10). Es debido a la N.

Si usas el subcomando "B", el cursor se desplaza hasta la posición que menciones, pero sin trazar ninguna línea. Detén la ejecución del programa y teclea lo siguiente

```
NEW
10 SCREEN 2
20 DRAW "BM80,80R10D10"
30 GOTO 30
RUN
```

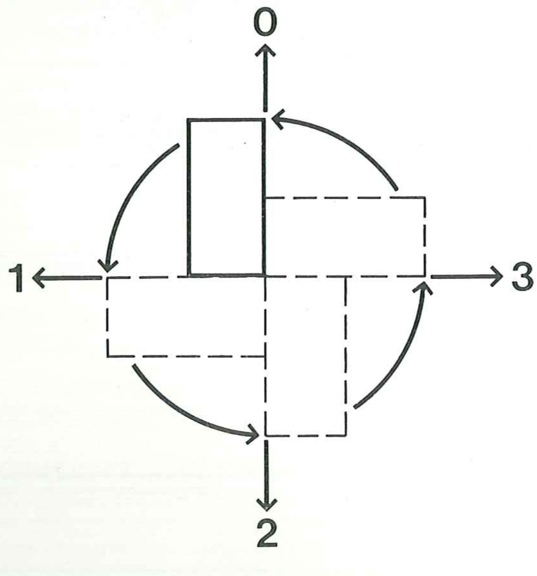
2. **Subcomandos de colores**

   Con el subcomando "Cn" puedes indicar el color con que deben efectuarse los trazos. El color es el correspondiente al parámetro N, de acuerdo con los números de colores.

3. **Subcomando para ángulo de orientación**

   Con el subcomando de clave "An" puedes decidir el ángulo con que se presentará en pantalla la figura trazada; y que corresponde a la dirección norte. El parámetro "n" es un número entre 0 y 3, de acuerdo con la siguiente 'inclinación':

   0 = 0 grados
   1 = 90 grados
   2 = 180 grados
   3 = 270 grados



4. **Subcomando para escala**

   El subcomando de clave "Sn" se usa para indicar al ordenador si debe ampliar o reducir la figura trazada. El parámetro "n" es un número entre 0 y 255, que se divide por 4 para conseguir el factor de escala real. El tamaño o longitud determinado por los subcomandos de claves "U", "D", "L", "R", "E", "F", "G", "H" y "M" es el tamaño originalmente establecido y que se multiplica por el factor de escala real. Así, por ejemplo, si en la instrucción de trazado incluimos "S8R10" se producirá una recta a la derecha, pero no con una longitud de 10 puntos de imagen como indica el subcomando "R10", sino con una longitud de 20 puntos de imagen (10 x 8 : 4 = 20), como resultado del subcomando "S8".

5. **Ejecutando subcomandos gráficos**

   Con MSX-BASIC todos los subcomandos pueden ser asignados como valor de una variable alfanumérica, de modo que puedas usarlo en combinación con la instrucción de trazado, mediante el subcomando de clave "X", que debe estar terminado por punto y coma. Teclea el siguiente ejemplo:

```
NEW
10 SCREEN 2
20 A$="E20R20G20L20"
30 DRAW "A0S4BM80,80XA$;"
40 GOTO 40
RUN
```

Este programa producirá en pantalla un rombo, cuyos 4 trazos vienen dados como valor de la variable alfanumérica "A$", que se incluye dentro de la instrucción de trazado usando el subcomando de clave "X". Un signo punto y coma (;) debe colocarse detrás del nombre de la variable.

Todas las constantes usadas en los subcomandos de una instrucción de trazado pueden sustituirse por variables, si se usa el signo igual delante del nombre de la variable y se termina dicho nombre por el signo punto y coma. Por ejemplo:

```
X1=10:X2=5
DRAW "M+=X1;,-=X2;"
```

La clave de instrucción "DRAW" para **trazar figuras,** tiene muchas posibilidades que debes ensayar por ti mismo. Aquí hay otro ejemplo. Detén la ejecución del programa (CTRL+STOP) y tecléalo:

```
NEW
10 SCREEN 2:COLOR 15,4,4
20 PSET (127,95),1
30 FOR I=1 TO 189 STEP 4
40 A$="L"+STR$(I)+"D"+STR$(I+1)+"R"+STR$(I+2)+"U"
   +STR$(I+3)
50 DRAW "S4XA$;"
60 NEXT I
70 CLS:GOTO 20
RUN
```

Solamente se puede parar la ejecución de este programa pulsando simultáneamente las teclas CTRL y STOP.

SPRITES

Capítulo 2.12

MSX-BASIC te permite crear figuras y hacer que sean llevadas a la pantalla en la posición que desees. Esas figuras se denominan también objetos móviles, y en la literatura inglesa se llaman 'sprites' que viene a ser algo así como duendes, genios, etc. Una de las propiedades de los 'sprites' es que se mueven en la parte frontal de la pantalla. Por ejemplo, la figura trazada en la parte del fondo de la pantalla, mediante las instrucciones de gráficas de trazado del capítulo anterior, queda intacta.

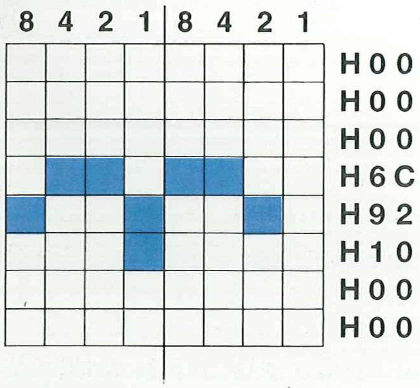
Para usar 'sprites', es necesario satisfacer dos requisitos:

1. El 'sprite' debe estar definido; lo que en otras palabras quiere decir que tú debes determinar la FORMA con que debe aparecer en pantalla.
2. Además, debes decidir el sitio en que dicho 'sprite' aparecerá en pantalla.

Definición del 'sprite'

Los 'sprites' pueden definirse en cuadrados de formato grande o pequeño. Un formato pequeño significa que el 'sprite' consta de 8 x 8 puntos de imagen. Usando 'sprites' pequeños, pueden definirse hasta un total de 256 'sprites'. La definición de cada 'sprite' se efectúa usando la variable del sistema cuya clave es "SPRITE$".

Supongamos que quieres definir el siguiente 'sprite':



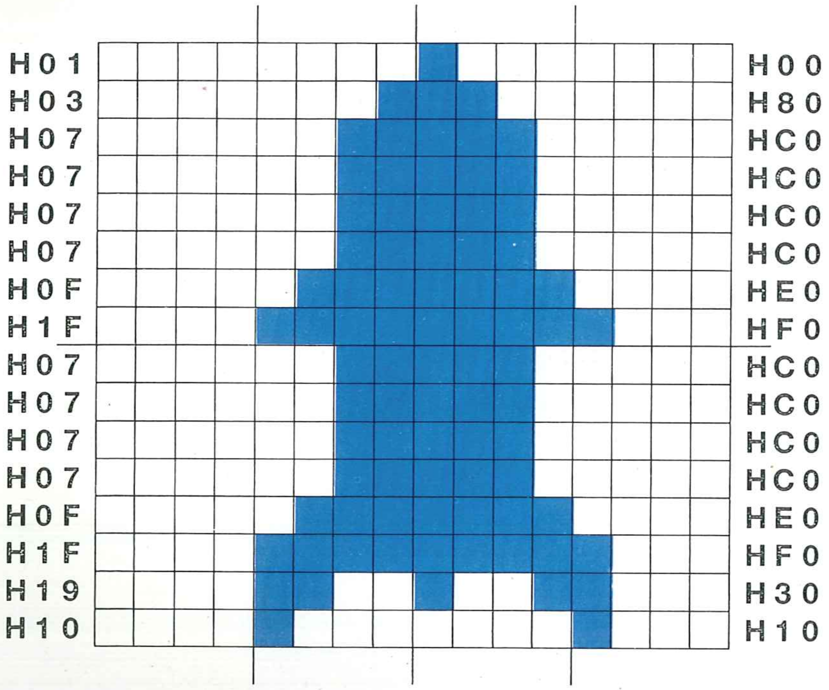
Véase también el Apéndice E.

Cada fila horizontal de 8 motas o puntos de imagen, forman un **octeto** que a su vez puede considerarse formado por 2 **cuartetos** (2 x 4 motas). Lo cómodo es definirlos usando la notación **hexadecimal,** tal y como se indica a la derecha de nuestro dibujo. El tamaño de los 'sprites' que pueden ser llevados a la pantalla, se debe especificar en la clave de instrucción "SCREEN", a continuación del parámetro que determina el modo de pantalla.

El 'sprite' del ejemplo, pudiera codificarse como:

```
10 SCREEN 2,0
20 A$=CHR$(&H00)+CHR$(&H00)+CHR$(&H00)+CHR$(&H6C)+
   CHR$(&H92)+CHR$(&H10)+CHR$(&H00)+CHR$(&H00)
30 SPRITE$(1)=A$
```

También pueden definirse los 'sprites' usando un tamaño mayor, en el que cada 'sprite' consta de 16 x 16 puntos de imagen, o motas. Cuando se usan 'sprites' grandes, sólo puedes definir 64 diferentes, usando la variable de clave "SPRITE$". Supongamos que quieres que aparezca en pantalla un 'sprite' como el indicado en este dibujo:



Cada fila horizontal de 16 motas forman 2 octetos, y cada uno puede a su vez dividirse en 2 cuartetos (2 x 4 motas). Se codifican usando la notación hexadecimal, pero en MSX-BASIC los octetos que definen el 'sprite' deben tomarse según una secuencia vertical. El 'sprite' de nuestro dibujo ejemplo, pudiera codificarse como sigue:

```
10 SCREEN 2,2
20 A$=CHR$(&H01)+CHR$(&H03)+CHR$(&H07)+CHR$(&H07)+
   CHR$(&H07)+CHR$(&H07)+CHR$(&H0F)+CHR$(&H1F)
30 B$=CHR$(&H07)+CHR$(&H07)+CHR$(&H07)+CHR$(&H07)+
   CHR$(&H0F)+CHR$(&H1F)+CHR$(&H19)+CHR$(&H10)
40 C$=CHR$(&H00)+CHR$(&H80)+CHR$(&HC0)+CHR$(&HC0)+
   CHR$(&HC0)+CHR$(&HC0)+CHR$(&HE0)+CHR$(&HF0)
50 D$=CHR$(&HC0)+CHR$(&HC0)+CHR$(&HC0)+CHR$(&HC0)+
   CHR$(&HE0)+CHR$(&HF0)+CHR$(&H30)+CHR$(&H10)
60 SPRITE$(1)=A$+B$+C$+D$
```

Los 'sprites' pueden también codificarse de una forma diferente, en cuanto a la notación empleada para los octetos. En nuestro ejemplo de 'sprite' pequeño, sería:

```
10 SCREEN 2,0
20 DATA 0,0,0,108,146,16,0,0
30 A$=""
40 FOR I=1 TO 8
50 READ A:A$=A$+CHR$(A)
60 NEXT I
70 SPRITE$(1)=A$
```

En este ejemplo, se usa la notación **decimal** equivalente a los 8 octetos que forman el 'sprite', y se incluyen en una instrucción "DATA". Estos valores son apuntados como valor de la variable "A" en la línea 50, y convertidos en alfanuméricos mediante la función de clave "CHR$", van empalmándose hasta formar el valor de la variable alfanumérica "A$".

Nuestro ejemplo de 'sprite' grande, pudiera codificarse según el mismo procedimiento como sigue:

```
10 SCREEN 2,2
20 DATA 1,3,7,7,7,7,15,31
30 DATA 7,7,7,7,15,31,25,16
40 DATA 0,128,192,192,192,192,224,240
50 DATA 192,192,192,192,224,240,48,16
60 A$=""
70 FOR I=1 TO 32
80 READ A:A$=A$+CHR$(A)
90 NEXT I
100 SPRITE$(1)=A$
```

En este ejemplo, los valores decimales equivalentes de los 32 octetos que componen el 'sprite', se reflejan en 4 instrucciones "DATA" consecutivamente. Luego, esos valores numéricos son apuntados en la línea 80 como valor de la variable numérica "A", y convertidos a alfanuméricos se van empalmando para formar el valor de la variable "A$".

Llevando los 'sprites' hasta la pantalla

En la clave de instrucción "SCREEN$" puede colocarse un parámetro inmediatamente a continuación del modo de pantalla para indicar las características de los 'sprites' que van a usarse. El valor de ese parámetro tiene el significado siguiente:

0 = 'sprites' pequeños, formato de 8 x 8
1 = 'sprites' pequeños, ampliados a formato (16 x 16)
2 = 'sprites' grandes, formato de 16 x 16
3 = 'sprites' grandes, ampliados a formato (32 x 32)

Por ejemplo, la siguiente instrucción:

```
10 SCREEN 2,3
```

Indica el modo 2 gráfico en pantalla, con 'sprites' de tamaño grande, ampliados a formato (32 x 32).

Hasta ahora no lo habíamos mencionado, pero ha quedado patente de lo anterior, que los 'sprites' llevados a pantalla pueden ampliarse hasta ocupar el doble del espacio originalmente definido para ellos.

Hay unas restricciones mínimas en relación con el uso de los 'sprites':

a. No pueden usarse en los modos 1 y 2 de texto.
b. No puedes usar combinaciones de diferentes tamaños de 'sprites'.
c. Sólo pueden llevarse simultáneamente a pantalla un máximo de 4 'sprites' sobre una misma fila horizontal.

Los 'sprites' se llevan a la pantalla mediante la clave de instrucción "PUT SPRITE", en la que se ha de incluir la siguiente información:

a. Número de prioridad.
b. La posición en la pantalla.
c. El color del 'sprite'.
d. El número del 'sprite'.

El número de prioridad, entre 0 y 31, además de identificar a cada uno de los 32 posibles 'sprites', establece la prioridad del mismo: a número menor, mayor prioridad; lo que significa lo siguiente:
Cuando 2 'sprites' son llevados a la misma posición de la pantalla, aparecerá en la parte frontal, por delante, el que tenga mayor prioridad.

La posición en la pantalla se indica como es habitual por las coordenadas X e Y. Pero en este caso, la coordenada X puede ser un número entre —32 y +255 y la coordenada Y un número entre —32 y +191. Es por tanto, posible llevar un 'sprite' fuera de la pantalla. Las coordenadas X e Y indican la malla superior e izquierda del 'sprite' definido.

Las coordenadas pueden darse como valores absolutos o valores relativos.

Cuando se da a la coordenada Y el valor de 209, el 'sprite' desaparece de la pantalla.

El color del 'sprite' está indicado por el parámetro numérico correspondiente al número de color (véase Apéndice C de este manual).

El número de 'sprite' es el parámetro final de la instrucción y determina cuál de los 'sprites' definidos previamente va a ser llevado sobre la pantalla del monitor.

Detén el programa (pulsando las teclas CTRL y STOP) y teclea el siguiente ejemplo:

```
NEW
10 SCREEN 2,0
20 DATA 0,3,31,63,255,255,63,7
30 DATA 0,128,248,255,255,254,252,192
40 DATA 0,0,0,108,146,16,0,0
50 FOR I=1 TO 3
60 A$=""
70 FOR J=1 TO 8
80 READ A:A$=A$+CHR$(A)
90 NEXT J
100 SPRITE$(I)=A$
110 NEXT I
120 PUT SPRITE 4,(180,50),15,1
130 PUT SPRITE 5,(188,50),15,2
140 FOR I=-32 TO 212
150 PUT SPRITE 3,(I,50),1,3
160 NEXT I
170 PUT SPRITE 3,(I,209)
180 FOR I=212 TO -32 STEP -1
190 PUT SPRITE 3,(I,50),1,3
200 NEXT I
210 GOTO 140
RUN
```

Si lo has tecleado adecuadamente, verás un pájaro negro pasando por delante de una nube blanca. La nube blanca se ha definido mediante dos 'sprites' (número 1 y 2) mientras el pájaro negro sólo ha necesitado usar el 'sprite' número 3.

La nube blanca es portada sobre la pantalla mediante las líneas 120 y 130 de programa, en posiciones adyacentes. El pájaro negro se mueve de izquierda a derecha, regido por el bucle "FOR... NEXT" establecido entre las líneas 140 a 160. El otro bucle "FOR... NEXT" establecido en las líneas 180 a 200, hace que el pájaro negro se mueva de derecha a izquierda. La instrucción 170 lo lleva fuera de la pantalla.

Detén ahora el programa y cambia la línea 10 para que sea:

```
10 SCREEN 2,1
```

Ahora ejecuta el nuevo programa y verás que el tamaño de los 'sprites' se ha doblado.

Vuelve a parar el programa y cambia la línea 190 para que sea:

```
190 PUT SPRITE 6,(I,50),1,3
```

Ahora, al ejecutar el programa de nuevo, verás que el pájaro continúa volando de izquierda a derecha, pero esta vez pasa por detrás de la nube en el camino de vuelta, ya que es llevado por un 'sprite' de menor prioridad (mayor número).

Colisiones de 'sprites'

MSX-BASIC también permite suscitar la ejecución de una subrutina predeterminada cuando dos 'sprites' cualesquiera colisionan entre sí. Esto se logra mediante la clave de instrucción "ON SPRITE GOSUB".

Detén el programa (pulsa las teclas CTRL y STOP) y cambia el programa anterior para que ahora sea:

```
5 SCREEN 2,0
10 ON SPRITE GOSUB 300
135 SPRITE ON
210 GOTO 135
300 PUT SPRITE 3,(I,209)
310 FOR J=50 TO 191
320 PUT SPRITE 3,(I,J),1,3
325 NEXT J
330 I=0:SPRITE OFF
340 RETURN 140
```

Ahora ejecuta este programa. La línea 10 de programa prepara el cepo que atrapará, detectará la colisión entre dos 'sprites', y señala al mismo tiempo la subrutina cuya ejecución se suscitará cuando ocurra tal percance. Pero es la línea 135 de programa la que arma, dispone dicho cepo para poder ser disparado. En la subrutina formada por las líneas 300 a 340, el pájaro es dirigido hacia abajo mediante un bucle «FOR... NEXT". En la línea 330, el cepo es desarmado, quitado; y la línea 340 instruye el ordenador para que vuelva no al lugar en que se desvió sino a la línea 140 del programa principal (que como no existe, pasa a la siguiente).

Puedes hacer que pare la ejecución pulsando simultáneamente las teclas CTRL y STOP, como ya estás acostumbrado.

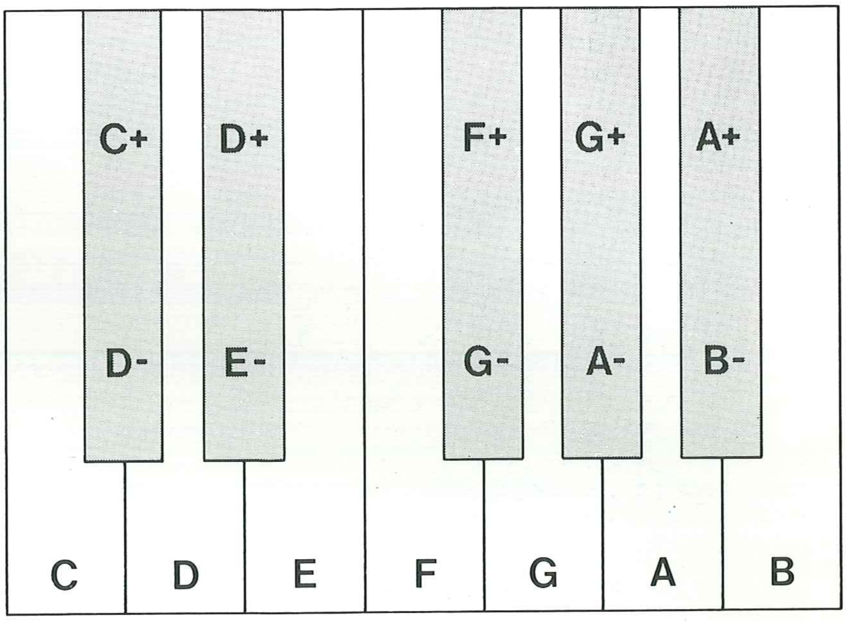
Música

Capítulo 2.13

En MSX-BASIC disponemos de una clave de instrucción "PLAY" que podemos usar cuando queremos que el ordenador ejecute alguna tonada musical. A continuación de esta clave, puedes indicar los **tonos** para tres voces, o CANALES de sonido. Las notas musicales emitidas por cada una de estas tres voces, se especifica mediante ciertas letras clave, subcomandos, de manera análoga a lo que se hace para trazar gráficos mediante la clave de instrucción "DRAW".

1. **Subcomandos para que suenen las notas de una octava**.

   Las letras clave de la A a la G, se usan para indicar las notas que han de sonar. Un signo mas (+) sube un semitono en la nota mencionada (A+ = A sostenido, agudo); mientras que un signo menos (—) produce un tono que es un semitono más bajo (A— = A bemol, grave).



Fundamentalmente, funciona como el teclado de un piano, lo que significa que el "+" solamente puede usarse cuando la tecla correspondiente del piano va seguida de una tecla negra, y un "—" sólo puede usarse cuando va precedida por una tecla negra. Por tanto, combinaciones tales como "F—" o "E+" no están permitidas.

Teclea por ejemplo, el siguiente comando de sonido:

```
PLAY "CEG"
```

2. **Subcomando para fijar la octava de la nota**

El tono de una nota se fija mediante el subcomando "On", siendo el parámetro "n" cualquier número del 1 al 8. Una **octava** estándar (como en el piano) va desde la nota "C" a la "B". Ejemplo

```
PLAY "O5CEGO4CEGO5CEG"
```

3. **Subcomando para tocar las notas**

En lugar de especificar una octava más una nota, es posible también señalar un **número de nota** con el subcomando "Nn", siendo el parámetro "n" cualquier número entre 0 y 96. Los dos ejemplos siguientes producen exactamente los mismos sonidos:

```
PLAY "O3CEG"
PLAY "N24N28N31"
```

4. **Subcomando de longitud de la nota (duración)**

El subcomando "Ln" se usa para determinar la longitud, la duración de una nota. El parámetro "n" es cualquier número entre 1 y 64, y tienen el siguiente significado:

1 = una nota completa (redonda).
2 = la mitad de una nota (blanca)
3 = un tercio de una nota.
4 = un cuarto de una nota (negra).
etc.
Ejemplo:

```
PLAY "L8DEL4DCE"
```

5. **Subcomando para pausa**

Una pausa o 'reposo', se especifica mediante el subcomando "Rn", siendo el parámetro "n" un número entre 1 y 64. Estos números representan las mismas longitudes o duraciones que en el caso del subcomando "Ln".

6. **Subcomando para cambiar la longitud de una nota**

Al poner un punto (.) detrás de la nota, se prolonga dicha nota hasta ser 1/2 de su longitud primitiva. Es también posible usar más de un punto detrás de una nota. Ejemplo:

```
PLAY "A..."
```

En este ejemplo, la nota "A" sonará durante 15/8 veces su longitud normal (=1 + 1/2 + 1/4 + 1/8).

7. **Subcomando de tempo**

El 'tempo' (número de notas por segundo), está determinado por el subcomando "Tn", siendo el parámetro "n" un número entre 32 y 255 que representa exactamente el número de 'negras' (cuartos de nota) durante un minuto.

8. **Subcomando de volumen**

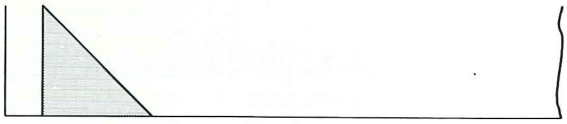
El volumen de la nota se especifica mediante el subcomando "Vn", siendo el parámetro "n" un número entre 0 y 15, que indica el volumen de menor a mayor. Ejemplo:

```
PLAY "V15CGR1V9CG"
```

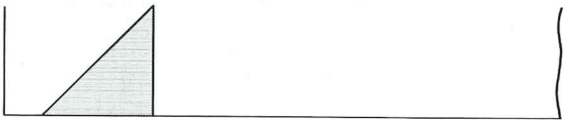
9. **Subcomando del perfil o forma del sonido**

El subcomando "Sn" se usa para especificar el perfil de la envolvente de una nota; siendo el parámetro "n" un número entre 0 y 15, que corresponde a los siguientes perfiles:

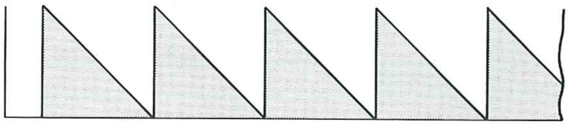
0 - 3 y 9



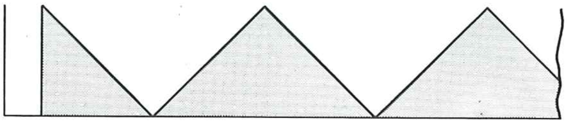
4 - 7 y 15



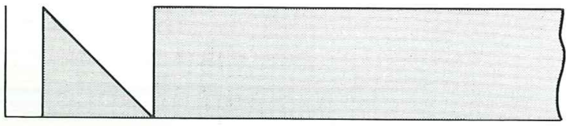
8



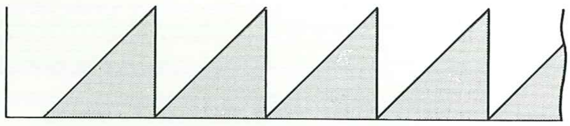
10



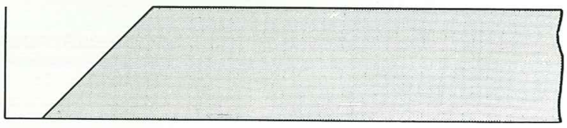
11



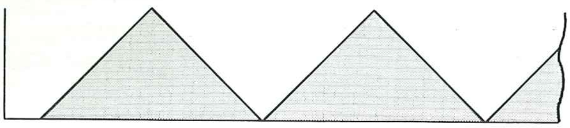
12



13



14

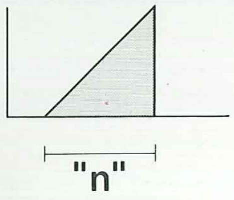


10. **Subcomando de longitud del perfil o forma**

"Mn" es el subcomando usado para elegir la longitud del perfil del sonido. El parámetro "n" puede ser cualquier número entre 1 y 65535. Simplemente escucha la diferencia en los siguientes ejemplos:

```
PLAY "S15M500C"
PLAY "S15M1000C"
PLAY "S12M500C"
PLAY "S12M1000C"
```

Cuanto más bajo sea el parámetro del subcomando "n", más breve es el perfil de la envolvente.



11. **Ejecutando subcomandos**

MSX-BASIC permite que los subcomandos de sonido sean asignados como valores de variables alfanuméricas y hacer mención de esa variable dentro de la instrucción PLAY, precediendo el nombre de la letra X y terminándolo con un punto y coma, como en el siguiente ejemplo:

```
NEW
10 A$="S12M1000C"
20 PLAY "XA$;"
RUN
```

Como puedes ver, debe colocarse inmediatamente detrás del nombre de la variable el signo punto y coma.

Todas las constantes incluidas en los subcomandos de la instrucción "PLAY" pueden sustituirse por variables numéricas, incluyendo el signo igual entre la letra clave del subcomando y el nombre de la variable y terminándola con punto y coma.

Por ejemplo:

```
X1=15
PLAY "N=X1;
```

Todos los subcomandos pueden usarse para los tres canales o voces de que está dotado el sistema. Un signo coma (,) debe colocarse para delimitar los subcomandos que corresponden a cada uno de los canales o voces. Por ejemplo:

```
PLAY "O4A","O5C","O3B"
```

La clave de instrucción "PLAY" ofrece muchas posibilidades que debes ensayar. Aquí hay otro ejemplo:

```
NEW
5 CLEAR 1000
10 A$="V8S1M9000T250L405"
20 B$="F+F+8E2R8EE8D2R8R4F+F+F+G2F+"
30 C$="04B05DD04B05D2E04B05D"
40 D$="04B.05DR4.B.A.DD1R2."
50 H$=B$+".R8"+B$+"04A"+C$+"DDD8C+2R404A"+C$+".D8D.
   D8E.R8R2"+B$+".R8"+D$
60 I$="R1R1R4AAAB2A.R8R1R1R4AAAB2AR1R1R1R1R1R1R1R1
   R1R1R4R8AAAB2A"
70 PLAY "XA$;","XA$;"
80 PLAY "XH$;","XI$;"
RUN
```

Puede que hayas reconocido esta melodía.

Con la clave de función "PLAY" (que no debe confundirse con la clave de instrucción "PLAY") puedes comprobar si una instrucción determinada de clave "PLAY" ha sido completamente ejecutada. Puedes ensayar con esto, añadiendo al programa anterior las siguientes líneas

```
85 CLS
90 PRINT PLAY(1):GOTO 90
```

Ahora ejecuta el programa y mientras está tocando la tonada, verás el valor —1 expuesto en la pantalla. Puedes parar la ejecución del programa pulsando simultáneamente las teclas CTRL y STOP.

Con la clave de instrucción "SOUND" también puede colocarse un valor específico en uno de los **registros** del circuito integrado, generador de sonidos; pero sólo debieras usar esta instrucción cuando estés familiarizado con la manera especial en que este circuito funciona.

MSX-BASIC y los JOYSTICKS para juegos

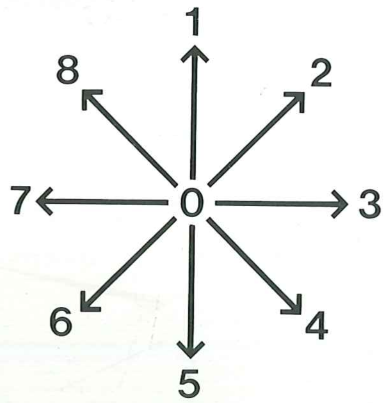
Capítulo 2.14

Puedes conectar uno o dos 'joysticks' a tu 'home computer' MSX. MSX-BASIC puede responder según la dirección en que muevas el mando. También puedes usar con el mismo fin, las teclas que controlan el movimiento del cursor. La dirección en que oprimes el mando del 'joystick', puede comprobarse usando la función de clave "STICK(n)", en la que el argumento "n" debe ser un número del 0 al 2 con el siguiente significado:

0 = Se usan las teclas de control de cursor como joysticks de juegos.
1 = Corresponde al joystick ajustado al conector 1.
2 = Corresponde al joystick ajustado al conector 2.

Los valores que resultan de aplicar la función "STICK(n)" con un determinado argumento, están comprendidos en la gama de 0 a 8, con el siguiente significado:

0 = Neutro (punto muerto).
1 = Norte.
2 = Nordeste.
3 = Este.
4 = Sudeste.
5 = Sur.
6 = Suroeste.
7 = Oeste.
8 = Noroeste.



Aquí hay un ejemplo:

```
NEW
10 A=STICK(0)
20 IF A=STICK(0) THEN 20
30 PRINT STICK(0)
40 GOTO 10
RUN
```

En este ejemplo, la línea 10 de programa hace que la variable numérica "A" sea igual al valor de las teclas de control del cursor. Mientras que ninguna de las teclas de cursor esté pulsada, el valor de dicha variable será por tanto, 0. Cuando se pulse una de dichas teclas pasará a ejecutarse la línea 30 exponiendo en pantalla el valor correspondiente. Ensaya con todas las teclas que controlan el movimiento del cursor.

MSX-BASIC puede también examinar si el botón accionador del joystck está pulsado. Con este mismo propósito se puede usar la barra espaciadora del teclado. Se efectúa mediante la clave de función "STRIG(n)", siendo el parámetro "n" un número de 0 a 4, que corresponde respectivamente a:

0 = la barra espaciadora es el botón accionador.
1 = Botón accionador 1 del joystck ajustado al conector 1.
2 = Botón accionador 1 del joystck ajustado al conector 2.
3 = Botón accionador 2 del joystck ajustado al conector 1.
4 = Botón accionador 2 del joystck ajustado al conector 2.

El valor obtenido como resultado de la función "STRIG(n)" es 0 cuando el botón accionador no está siendo oprimido y —1 cuando sí está apretado.

Ejemplo:

```
NEW
10 IF STRIG(0)=-1 THEN PRINT STRIG(0)
20 GOTO 10
RUN
```

Por la línea 10, MSX-BASIC comprueba si se ha pulsado la barra espaciadora, y si está pulsada expone el valor resultado de la función "STRIG(0)".

Lanzando un cohete

En el siguiente ejemplo se lanza un cohete al pulsar la barra espaciadora. Teclea el programa siguiente con cuidado y procura no cometer equivocaciones

```
NEW
10 SCREEN 2,3:COLOR 1,4,1
20 DATA 0,0,0,0,3,7,15,63
30 DATA 63,127,127,63,31,1,0,0
40 DATA 0,0,0,0,192,252,255,255
50 DATA 255,255,255,255,255,255,28,0
60 DATA 0,0,0,1,15,31,255,255
70 DATA 255,255,255,255,255,255,30,0
80 DATA 0,0,0,128,224,248,248,252
90 DATA 252,248,248,240,240,224,0,0
100 DATA 0,0,0,28,126,127,255,127
110 DATA 31,15,7,3,0,0,0,0
120 DATA 0,0,0,0,96,240,248,252
130 DATA 254,254,252,252,248,0,0,0
140 DATA 0,0,0,0,0,0,3,15
150 DATA 63,255,255,127,63,31,0,0
160 DATA 0,0,0,0,0,248,252,254
170 DATA 254,252,252,252,252,248,248,0
180 DATA 0,0,0,0,0,0,0,0
190 DATA 0,0,0,0,0,0,0,0
200 DATA 3,3,7,7,7,7,7,15
210 DATA 15,15,15,31,31,31,31,31
220 DATA 0,0,0,0,0,0,0,0
230 DATA 0,0,0,0,0,0,0,0
240 DATA 255,255,255,16,40,16,0,0
250 DATA 0,0,0,0,0,0,0,0
```

```
260 DATA 255,255,255,0,1,0,0,0
270 DATA 0,0,0,0,0,0,0,0
280 DATA 240,240,240,128,64,128,0,0
290 DATA 0,0,0,0,0,0,0,0
300 DATA 8,28,28,62,62,62,127,255
310 DATA 62,62,62,62,127,255,201,128
320 DATA 0,0,0,0,0,0,0,128
330 DATA 0,0,0,0,0,128,128,128
340 DATA 8,28,28,62,62,62,62,28
350 DATA 28,20,0,0,0,0,0,0
360 DATA 0,0,0,0,0,0,0,0
370 DATA 0,0,0,0,0,0,0,0
380 FOR J=1 TO 9:A$=""
390 FOR I=1 TO 32:READ A:A$=A$+CHR$(A):NEXT I
400 SPRITE$(J)=A$:NEXT J
410 CIRCLE (88,120),16,6,-0.0001,-3.415,0.5
420 PAINT (88,118),6
430 LINE (0,192)-(255,120),12,BF
440 PUT SPRITE 1,(80,16),15,1
450 PUT SPRITE 2,(112,16),15,2
460 PUT SPRITE 10,(160,80),15,3
470 PUT SPRITE 11,(16,48),15,4
480 I=120:J=144:GOSUB 1000
490 IF STRIG(0)=-1 GOTO 2000
500 IF STICK(0)=3 THEN GOSUB 3000
510 IF STICK(0)=7 THEN GOSUB 4000
520 GOTO 490
1000 PUT SPRITE 5,(I,J),1,5:J=J+32
1010 PUT SPRITE 6,(I,J),1,6:I=I+32
1020 PUT SPRITE 7,(I,J),1,7:J=J-32
1030 PUT SPRITE 8,(I,J),7,8:I=I-32
1040 RETURN
2000 I=I+32:FOR K=144 TO -33 STEP -1
2010 PUT SPRITE 8,(I,K),7,8:M=K+32
2020 PUT SPRITE 9,(I,M),9,9
2025 FOR T=I TO 1:NEXT T
2030 PUT SPRITE 9,(I,209)
2040 NEXT K:FOR K=1 TO 1000:NEXT K:COLOR 15,4,7:END
3000 I=I+1:IF I>200 THEN I=200
3010 GOSUB 1000:RETURN
4000 I=I-1:IF I<0 THEN I=0
4010 GOSUB 1000:RETURN
```

Ahora ejecuta este programa. Puedes desplazar la plataforma de lanzamiento a la derecha o a la izquierda con las teclas de control del cursor. El cohete se lanza cuando oprimes la barra espaciadora. Si el programa no funciona apropiadamente, comprueba que todas las instrucciones han sido tecleadas correctamente.

HARDWARE MSX

Capítulo 3

3.1. Mantenimiento
3.2. Periféricos.

Mantenimiento

Capítulo 3.1

El 'home computer' puede limpiarse con un paño seco. No uses nunca detergentes químicos.

Si usas una lecto-grabadora de datos, las cabezas de grabación y lectura deberán limpiarse regularmente, de acuerdo con las instrucciones suministradas por el fabricante.

Asegúrate que conservas tus cassettes en lugares frescos, manteniéndolos alejados de la luz directa del sol y de otras fuentes de calor y aislados de campos magnéticos. Si no lo haces así, puede que pierdas la información grabada.

Si tu 'home computer' no funciona apropiadamente (si oyes un ruido extraño o si produce un olor raro, por ejemplo, debido a sobrecalentamiento), apaga inmediatamente el ordenador y llévalo a tu distribuidor para que verifique su funcionamiento.

Todas las reparaciones deberán efectuarse por el distribuidor o a través de él y tú nunca deberás abrir la consola por tí mismo.

Cosas que no debes hacer

Como todos los otros equipos electrónicos, tu 'home computer' es alérgico a la excesiva humedad. Asegúrate que nunca derramas líquidos sobre tu 'home computer'.

Las aberturas de ventilación están previstas para la refrigeración. Asegúrate que están siempre libres de obstáculos, con el fin de mantener una apropiada circulación de aire.

Mantén el 'home computer' alejado de fuentes de calor, tales como estufas, radiadores o luz directa del sol.

No toques nunca las patillas y contactos de los conectores con tus dedos, dado que puede causar corrosión y engrasamiento.

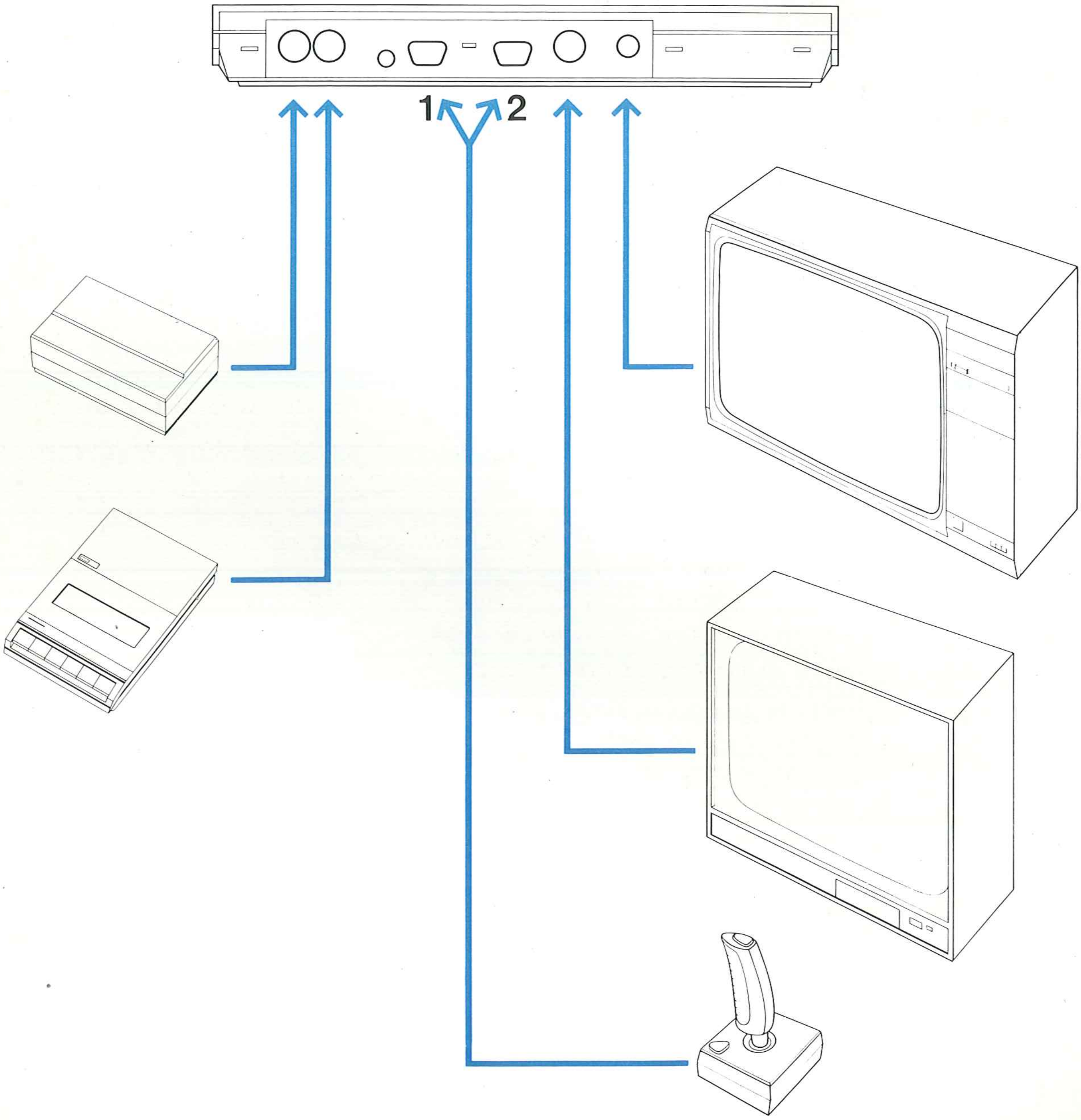
Coloca los cables de conexión, de manera que nadie pueda tropezar o enredarse con ellos.

Al quitar una clavija, no tires del cable, sino agarra firmemente de la propia clavija.

No dejes caer tu 'home computer' cuando lo muevas de un lado a otro, ni pongas cosas pesadas encima de la consola.

Periféricos

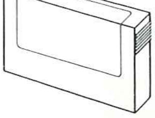
Capítulo 3.2 En el panel trasero de tu 'home computer' encontrarás conectores estándar para los siguientes equipos periféricos y accesorios:



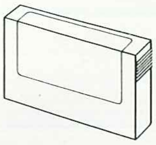
Nota. También puede ser usado otro equipo (véanse las instrucciones operativas suministradas con los periféricos pertinentes).

Entre otros, se pueden conectar los siguientes periféricos a través de las ranuras para cartuchos:

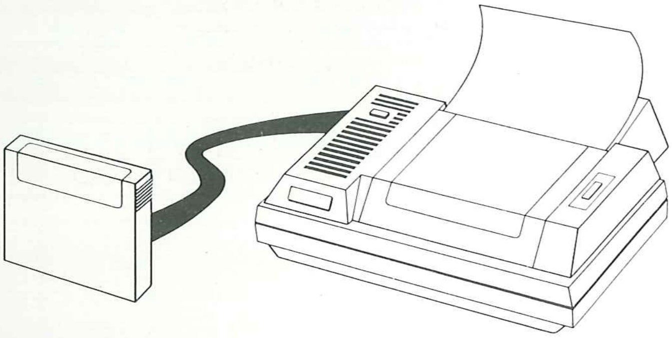
Cartucho de memoria de 16K RAM.



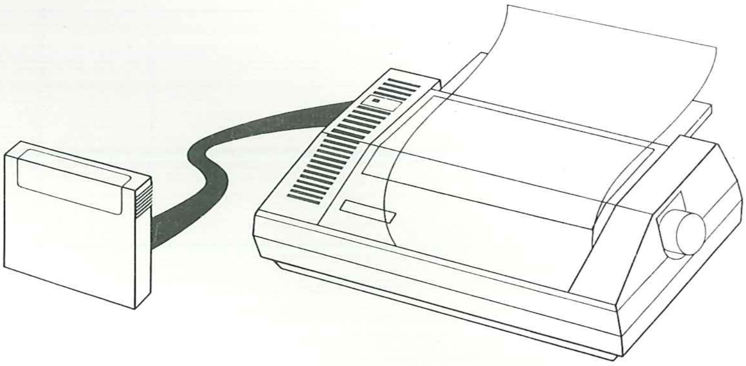
Cartucho de programas en ROM.



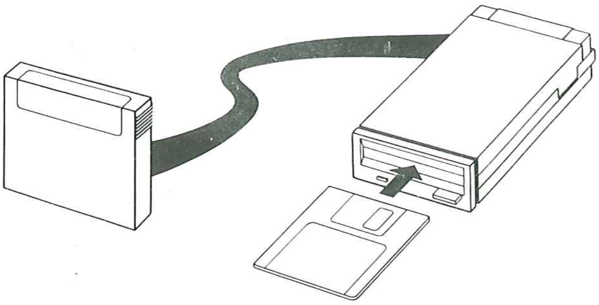
Impresora de 40 columnas.



Impresora de 80 columnas.



Unidad de diskette.



Apéndices

Capítulo 4

Lista de mensajes de error

Apéndice A

MENSAJE DE ERROR	CODIGO	EXPLICACION
"Bad file name"	56	Se ha usado un nombre de fichero incorrecto.
"Bad file number"	52	El número de fichero mencionado corresponde a uno que todavía no ha sido abierto mediante la instrucción "OPEN", o el número de fichero es mayor que el especificado en la instrucción "MAXFILES".
"Can't continue"	17	El operador está intentando que SIGA la ejecución de un programa que: — se ha interrumpido debido a un mensaje de error — se ha alterado o cambiado mientras la ejecución estaba interrumpida — no existe.
"Device I/O error"	19	Se ha observado un error en un equipo de entrada o salida, mientras se depositaba o recuperaba un fichero o un programa.
"Direct statement"	57	Se ha encontrado una frase directa durante el proceso de carga de un fichero ASCII.
"Division by zero"	11	Mientras se efectúa un cálculo, el ordenador encuentra una división por 0 o una situación que le obliga a elevar 0 a una potencia negativa.
"Field overflow"	50	Se REBASA el número total de octetos permitidos en la instrucción "FIELD", cuyo máximo es de 256.
"File already open"	54	Se pide al ordenador que abra un fichero mediante la instrucción "OPEN", y ese fichero **ya ha sido abierto.** O se ha dado un comando "KILL" para eliminar un fichero que todavía está abierto.
"File not open"	59	El operador menciona un fichero en una instrucción de lectura o de escritura, y el fichero todavía no ha sido abierto con la instrucción "OPEN".
"Illegal direct"	12	Se ha usado en el modo directo como comando, una clave de instrucción que no es legal en ese modo.

"Illegal function call"	5	Se especifica un parámetro inadecuado al citar una función. Puede ocurrir como resultado de: — un sufijo o un subíndice negativo o irrazonablemente grande — un argumento 0 ó negativo con la función logarítmica — un argumento negativo con la función raíz cuadrada — un argumento incorrecto con alguna de las claves "MID$", "LEFT$", "RIGHT$", "INP", "OUT", "PEEK", "POKE", "TAB", "SPC", "STRING$", "SPACE$", "INSTR$", "ON... GOTO", "ON... GOSUB".
"Input past end"	55	El operador está intentando imponer los datos tomados de un fichero, y ya se ha pasado el final del mismo. Usa la clave de función "EOF" para prevenir este error.
"Internal error"	51	Se ha producido una situación inesperada que no puede ser manejada por el interpretador BASIC.
"Line buffer overflow"	25	El operador ha REBASADO el buffer de línea, al hacerle contener demasiados caracteres.
"Missed operand"	24	A una expresión que contiene una función, instrucción o comando, le FALTA un parámetro.
"NEXT without FOR"	1	La variable mencionada en la clave de instrucción "NEXT" no es la misma que la variable en la clave de instrucción correspondiente "FOR"; o se encuentra una clave de instrucción "'NEXT" sin que hubiera la precedente clave de instrucción "FOR".
"No RESUME"	21	Una subrutina suscitada por la aparición de un error, no tiene instrucción "RESUME".
"Out of DATA"	4	Se intenta apuntar un dato de una serie expresada en instrucciones "DATA", mediante la clave de instrucción "READ" y no queda ninguna constante sin apuntar.
"Out of memory"	7	El programa es demasiado grande, o contiene demasiados bucles, desvíos a subrutinas o variables alfanuméricas. Puede también suceder cuando contiene expresiones demasiado complicadas.
"Out of string space"	14	Las variables alfanuméricas exceden del espacio disponible para las **cadenas** de caracteres. Usa la clave de instrucción "CLEAR" para generar espacio adicional.
"Overflow"	6	El resultado de un cálculo rebasa o sobrepasa el tamaño o precisión permitida.
"Redimensional array"	10	Se usan dos claves de instrucción "DIM" con el mismo nombre para la variable múltiple o tabla. Consulta la clave de instrucción "ERASE".
"RESUME without error"	22	Se pide al ordenador que reanude una instrucción "RESUME" cuando solamente puede hacerse al suscitarse una subrutina de tratamiento de errores. ("ON ERROR GOTO").

"RETURN without GOSUB"	3	Se encuentra una clave de instrucción "RETURN" para la que no ha existido ninguna clave de instrucción previa "GOSUB".
"Sequential I/O only"	58	El operador está intentando leer o escribir "referencialmente" con un fichero únicamente secuencial.
"String formula too complex"	16	La expresión alfanumérica (cadenas de caracteres) es demasiado compleja o demasiado grande.
"String too long"	15	Se intenta incluir más de 255 caracteres en una variable alfanumérica (cadena de caracteres).
"Subscript out of range"	9	Se intenta mencionar un elemento de una tabla con un subíndice (sufijo) que está fuera de la gama correspondiente.
"Syntax error"	2	Una clave de comando, de instrucción o de función, está deletreada incorrectamente o la frase presenta una forma sintáctica no admitida (faltan signos de puntuación, etc.).
"Type mismatch"	13	Hay **discordancia** en el tipo dado a la variable y el tipo que corresponde al valor de la misma. Puede ser la variable alfanumérica y el valor numérico o viceversa.
"Unidentified line number"	8	Se ha dado un comando, o una instrucción, que menciona una línea de programa cuyo número no existe.
"Unidentified user function"	18	Se cita una función de usuario que no ha sido definida previamente mediante la clave de instrucción "DEF".
"Unprintable error"	23 26 a 49 60 a 255	Estos códigos no tienen asociado ningún mensaje de error, por lo que el error es 'no-mostrable'. El usuario puede usar estos códigos para definir sus propios mensajes de error.
"Verify error"	20	Se produce al verificar el programa contenido en la memoria con el programa que está grabado en el cassette, después de haber ejecutado la instrucción "CLOAD?".

Funciones matemáticas

Apéndice B

Las funciones matemáticas que no están incluidas intrínsecamente en MSX-BASIC, pueden derivarse según las fórmulas siguientes:

Funciones derivadas	**Fórmula en MSX-BASIC**
SECANTE	= 1/COS(X)
COSECANTE	= 1/SIN(X)
COTANGENTE	= 1/TAN(X)
ARCO SENO	= ATN(X/SQR(—X*X+1))
ARCO COSENO	= —ATN(X/SQR(—X*X+1))+1.5708
ARCO SECANTE	= ATN(X/SQR(X*X—1))+SGN(SGN(X)—1)*1.5708)
ARCO COSECANTE	= ANT(X/SQR(X*X—1))+(SGN(X)—1)*1.5708
ARCO COTANGENTE	= ANT(X)+1.5708
SENO HIPERBOLICO	= (EXP(X)—EXP(—X))/2
COSENO HIPERBOLICO	= (EXP(X)+EXP(—X))/2
TANGENTE HIPERBOLICA	= EXP(X)/EXP(X)+EXP(—X)*2+1
SECANTE HIPERBOLICA	= 2/(EXP(X)+EXP(—X))
COSECANTE HIPERBOLICA	= 2/(EXP(X)—EXP(—X))
COTANGENTE HIPERBOLICA	= EXP(—X)/EXP(X)+EXP(—X)*2+1
ARCO SENO HIPERBOLICO	= LOG(X+SQR(X*X+1))
ARCO COSENO HIPERBOLICO	= LOG(X+SQR(X*X—1))
ARCO TANGENTE HIPERBOLICA	= LOG((1+X)/1—X))/2
ARCO SECANTE HIPERBOLICA	= LOG((SQR(—X*X+1)+1)/X)
ARCO COSECANTE HIPERBOLICA	= LOG((SGN(X)*SQR(X*X+1)+1)/X)
ARCO COTANGENTE HIPERBOLICA	= LOG((X+1)/(X—1))/2

Tabla de colores

Apéndice C

Número de color	Color
0	Transparente
1	Negro
2	Verde
3	Verde pálido
4	Azul oscuro
5	Azul pálido
6	Rojo oscuro
7	Ciano
8	Rojo
9	Rojo pálido
10	Amarillo oscuro
11	Amarillo pálido
12	Verde oscuro
13	Magenta
14	Gris
15	Blanco

Funciones de control

Apéndice D

En la siguiente tabla encontrarás un resumen de todas las funciones de control en MSX-BASIC. Estas funciones se ejecutan al pulsar simultáneamente la tecla CTRL y la otra tecla mencionada.

Cada función de control tiene asociado intrínsecamente un código, que puede usarse a discreción, como por ejemplo, como argumento de la función CHR$.

```
PRINT CHR$(7)
```

Produce un breve pitido.

Al final de la tabla encontrarás una breve descripción de las diversas funciones de control.

CODIGO	CTRL + tecla:	TECLA ESPECIAL	FUNCION
1	A		Caracteres alternativos
2	B		Cursor a palabra precedente
3	C		Detiene el comando AUTO
4	D		Ninguna
5	E		Trunca línea de texto
6	F		Cursor a palabra siguiente
7	G		Pitido
8	H	BS	Retroceso (backspace)
9	I	TAB	Tabulador
10	J		Avance de línea
11	K	HOME	Cursor a posición **base**
12	L	CLR	Limpia pantalla
13	M	RETURN	Retorno de carro
14	N		Cursor a final de línea
15	O		Ninguna
16	P		Ninguna
17	Q		Ninguna
18	R	INS	Inserción
19	S		Ninguna
20	T		Ninguna
21	U		Borra línea
22	V		Ninguna
23	W		Ninguna
24	X	SELECT	Ninguna
25	Y		Ninguna
26	Z		Ninguna
27	[	ESC	Ninguna
28	\	→	Cursor a la derecha
29	]	←	Cursor a la izquierda
30	^	↑	Cursor arriba
31	—	↓	Cursor abajo

Descripción de las funciones de control

Caracteres alternativos
Con esta función, el siguiente carácter será del repertorio alternativo (véase Apéndice E).

Cursor a palabra precedente
El cursor se desplazará situándose sobre el primer carácter de la palabra anterior. Para MSX-BASIC el primer carácter es de la "A" a la "Z", de la "a", a la "z", o del 0 al 9.

Detiene el comando AUTO
Esta función de control vuelve a MSX-BASIC a nivel de comando si se había ejecutado el comando de numeración AUTOMATICA.

Trunca línea de texto
Se quitan todos los símbolos situados desde la posición presente del cursor hasta el final de esa línea.

Cursor a palabra siguiente
El cursor se desplaza situándose en el primer carácter de la siguiente palabra. Para MSX-BASIC el primer carácter es de la "A" a la "Z", de la "a", a la "z", o del 0 al 9

Pitido
Esta función genera un breve pitido.

Retroceso (BS)
Esta función de control QUITA el carácter situado inmediatamente a la izquierda de la posición presente del cursor; y desplaza —retrocede— el propio cursor y todos los caracteres situados a su derecha.

Tabulador
El cursor se desplaza hasta el siguiente tope de tabulación. Hay un tope de tabulación cada 8 posiciones de pantalla.

Avance de línea
El cursor **avanza,** se desplaza hasta la primera posición de la siguiente línea de texto.

Cursor a posición base
El cursor se mueve a la posición base, i.e., la esquina superior izquierda de la pantalla, sin dejar la pantalla en blanco.

Limpiar pantalla
El cursor se desplaza a la posición base, al mismo tiempo que se **limpia** la pantalla.

Retorno de carro
Marca el final de una línea de programa, y al mismo tiempo introduce lo tecleado para que sea interpretado por MSX-BASIC.

Cursor a final de línea
El cursor se sitúa sobre la primera posición que vaya inmediatamente detrás del último carácter de esa línea.

Inserción
Permite insertar, incluir caracteres, entre los expuestos en pantalla. Si esta función de contol se repite, los nuevos caracteres sustituirán a los caracteres ya existentes en pantalla.

Borra línea
Se elimina completamente la línea donde está situado el cursor.

Cursor a la derecha
Se desplaza una posición a la derecha.

Cursor a la izquierda
Se desplaza una posición a la izquierda.

Cursor arriba
El cursor asciende una línea.

Cursor abajo
El cursor desciende una línea.

Repertorio de caracteres

Apéndice E

En este apéndice encontrarás todos los **caracteres** que pueden usarse con MSX-BASIC, junto con sus correspondientes **códigos asociados.** Estos códigos pueden usarse, por ejemplo, como argumento de la función de clave CHR$.

Los siguientes dos ejemplos, exponen en pantalla el mismo carácter:

```
PRINT "A"
```

o bien

```
PRINT CHR$(65)
```

En el modo 1 de texto (40 caracteres por fila o renglón) las dos columnas del extremo derecho de la imagen, que componen la matriz que da forma a los caracteres, no aparece en pantalla.

Ensaya el siguiente ejemplo:

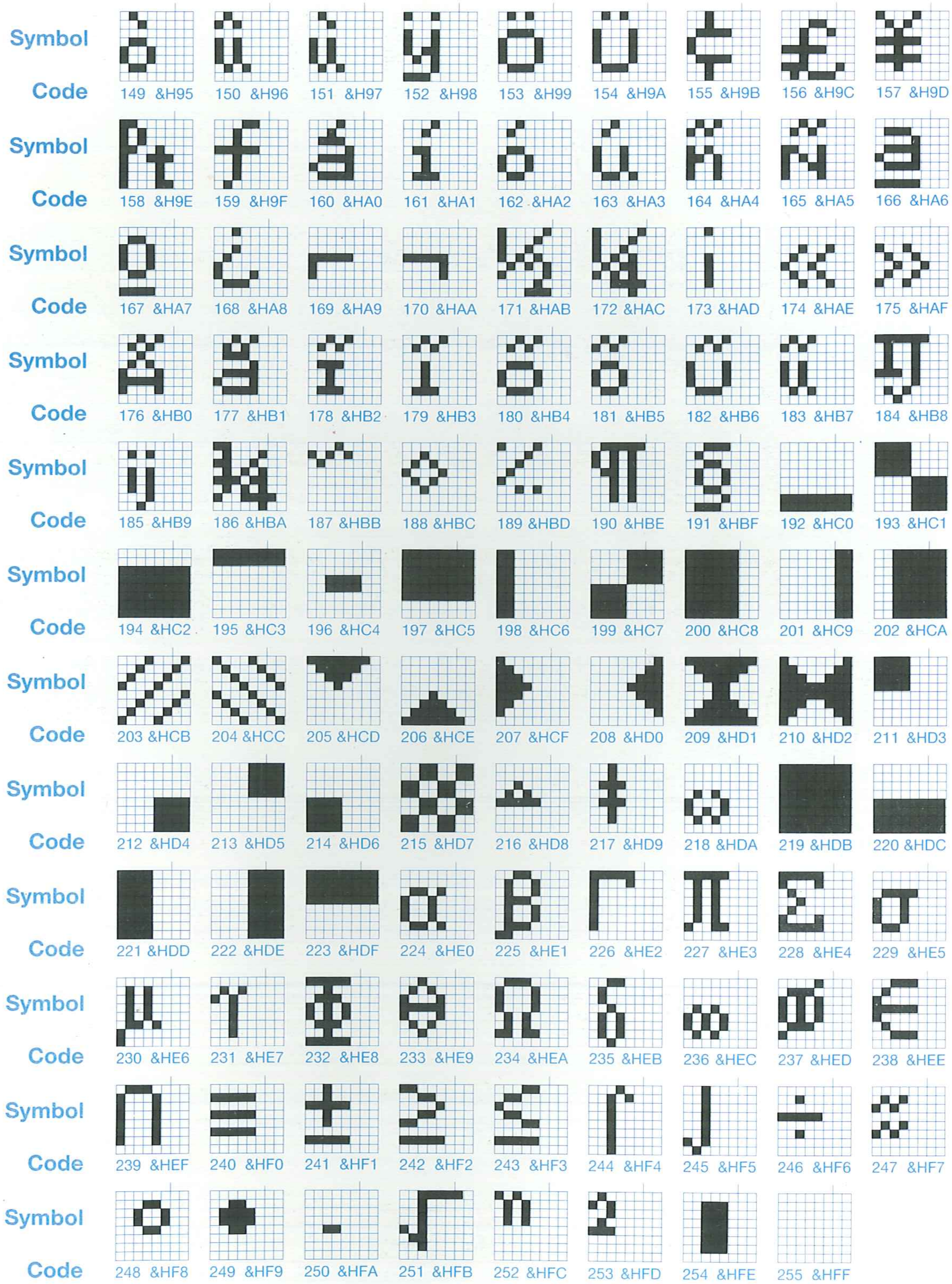
```
NEW
10 SCREEN 0
20 PRINT CHR$(210)
RUN
```

Ahora cambia la línea 10 para que sea:

```
10 SCREEN 1
```

Ahora ejecuta de nuevo el programa y observarás la diferencia.

Symbol		!	"	#	$	%	&	'	(
Code	32 &H20	33 &H21	34 &H22	35 &H23	36 &H24	37 &H25	38 &H26	39 &H27	40 &H28
Symbol	)	*	+	,	-	.	/	0	1
Code	41 &H29	42 &H2A	43 &H2B	44 &H2C	45 &H2D	46 &H2E	47 &H2F	48 &H30	49 &H31
Symbol	2	3	4	5	6	7	8	9	:
Code	50 &H32	51 &H33	52 &H34	53 &H35	54 &H36	55 &H37	56 &H38	57 &H39	58 &H3A
Symbol	;	<	=	>	?	@	A	B	C
Code	59 &H3B	60 &H3C	61 &H3D	62 &H3E	63 &H3F	64 &H40	65 &H41	66 &H42	67 &H43
Symbol	D	E	F	G	H	I	J	K	L
Code	68 &H44	69 &H45	70 &H46	71 &H47	72 &H48	73 &H49	74 &H4A	75 &H4B	76 &H4C
Symbol	M	N	O	P	Q	R	S	T	U
Code	77 &H4D	78 &H4E	79 &H4F	80 &H50	81 &H51	82 &H52	83 &H53	84 &H54	85 &H55
Symbol	V	W	X	Y	Z	[	\	]	^
Code	86 &H56	87 &H57	88 &H58	89 &H59	90 &H5A	91 &H5B	92 &H5C	93 &H5D	94 &H5E
Symbol	_	`	a	b	c	d	e	f	g
Code	95 &H5F	96 &H60	97 &H61	98 &H62	99 &H63	100 &H64	101 &H65	102 &H66	103 &H67
Symbol	h	i	j	k	l	m	n	o	p
Code	104 &H68	105 &H69	106 &H6A	107 &H6B	108 &H6C	109 &H6D	110 &H6E	111 &H6F	112 &H70
Symbol	q	r	s	t	u	v	w	x	y
Code	113 &H71	114 &H72	115 &H73	116 &H74	117 &H75	118 &H76	119 &H77	120 &H78	121 &H79
Symbol	z	{	|	}	~		Ç	ü	é
Code	122 &H7A	123 &H7B	124 &H7C	125 &H7D	126 &H7E	127 &H7F	128 &H80	129 &H81	130 &H82
Symbol	â	ä	à	å	ç	ê	ë	è	ï
Code	131 &H83	132 &H84	133 &H85	134 &H86	135 &H87	136 &H88	137 &H89	138 &H8A	139 &H8B
Symbol	î	ì	Ä	Å	É	æ	Æ	ô	ö
Code	140 &H8C	141 &H8D	142 &H8E	143 &H8F	144 &H90	145 &H91	146 &H92	147 &H93	148 &H94


Symbol
Code
149 &H95
150 &H96
151 &H97
152 &H98
153 &H99
154 &H9A
155 &H9B
156 &H9C
157 &H9D
158 &H9E
159 &H9F
160 &HA0
161 &HA1
162 &HA2
163 &HA3
164 &HA4
165 &HA5
166 &HA6
167 &HA7
168 &HA8
169 &HA9
170 &HAA
171 &HAB
172 &HAC
173 &HAD
174 &HAE
175 &HAF
176 &HB0
177 &HB1
178 &HB2
179 &HB3
180 &HB4
181 &HB5
182 &HB6
183 &HB7
184 &HB8
185 &HB9
186 &HBA
187 &HBB
188 &HBC
189 &HBD
190 &HBE
191 &HBF
192 &HC0
193 &HC1
194 &HC2
195 &HC3
196 &HC4
197 &HC5
198 &HC6
199 &HC7
200 &HC8
201 &HC9
202 &HCA
203 &HCB
204 &HCC
205 &HCD
206 &HCE
207 &HCF
208 &HD0
209 &HD1
210 &HD2
211 &HD3
212 &HD4
213 &HD5
214 &HD6
215 &HD7
216 &HD8
217 &HD9
218 &HDA
219 &HDB
220 &HDC
221 &HDD
222 &HDE
223 &HDF
224 &HE0
225 &HE1
226 &HE2
227 &HE3
228 &HE4
229 &HE5
230 &HE6
231 &HE7
232 &HE8
233 &HE9
234 &HEA
235 &HEB
236 &HEC
237 &HED
238 &HEE
239 &HEF
240 &HF0
241 &HF1
242 &HF2
243 &HF3
244 &HF4
245 &HF5
246 &HF6
247 &HF7
248 &HF8
249 &HF9
250 &HFA
251 &HFB
252 &HFC
253 &HFD
254 &HFE
255 &HFF

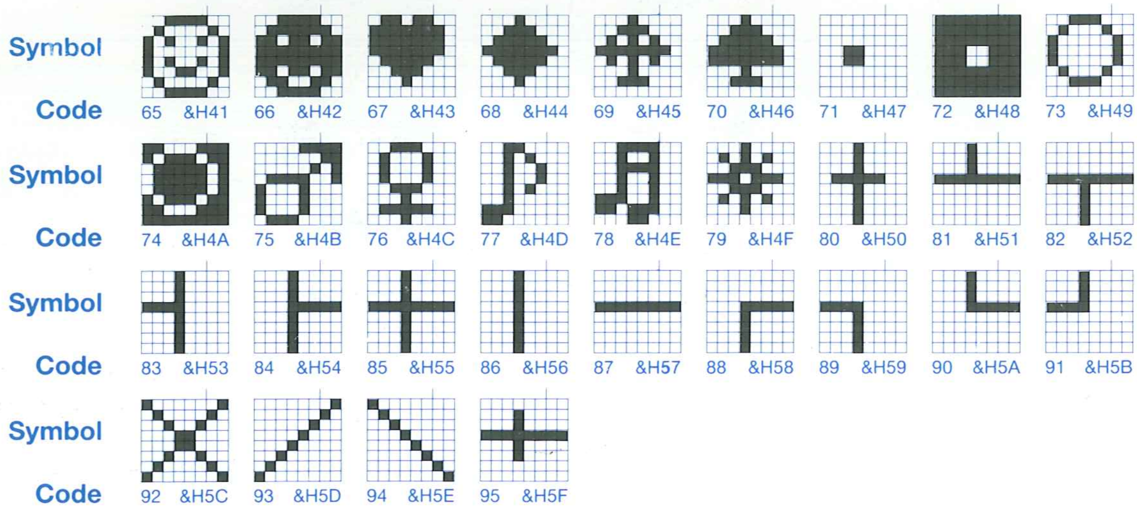
Caracteres alternativos

Además de los anteriores **símbolos,** hay otros cuantos símbolos que tienen asociados los **códigos** 65 a 95, y se consiguen precediéndolos de la función "CHR$(1)".

Ejemplo:

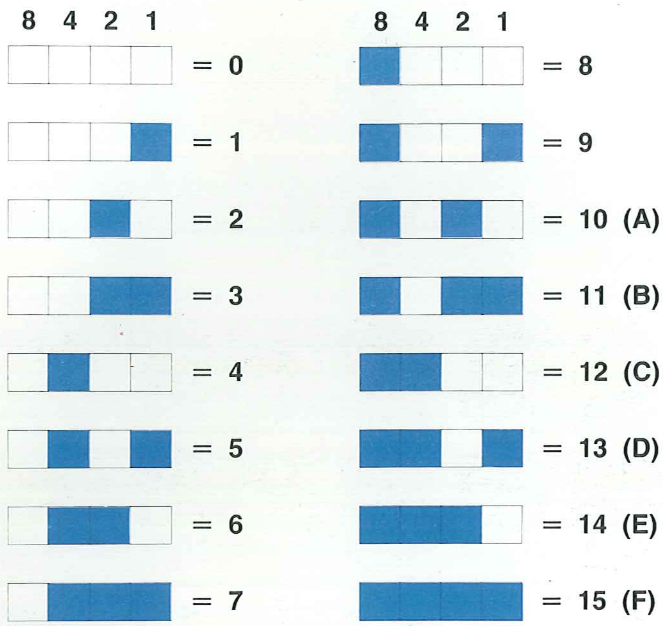
```
NEW
10 SCREEN 1
20 PRINT CHR$(65)
30 PRINT CHR$(1)+CHR$(65)
RUN
```

La línea 20 producirá el carácter estándar o regular (="A"), mientras que la línea 30 producirá el carácter **alternativo** correspondiente al código 65 (= una cara con ojos y boca).



Matriz de caracteres

Cada línea de la matriz que da forma a un carácter es un octeto que puede subdividirse en dos cuartetos (2 x 4 bits). Los cuatro bits tienen asignados respectivamente, de izquierda a derecha los valores decimales de 8, 4, 2 y 1; cuando se desea calcular su equivalente decimal. Con esa notación, 4 bits contiguos y examinados simultáneamente pueden tener un valor entre 0 y 15:



Usando la notación hexadecimal, los valores decimales 10 a 15, vienen sustituidos por los símbolos hexadecimales A a F. El mínimo valor por línea en notación hexadecimal es pues de 00, y el máximo valor en dicha notación es de FF.

Palabras reservadas (CLAVE)

Apéndice F

Las siguientes palabras no pueden usarse como nombres de variables:

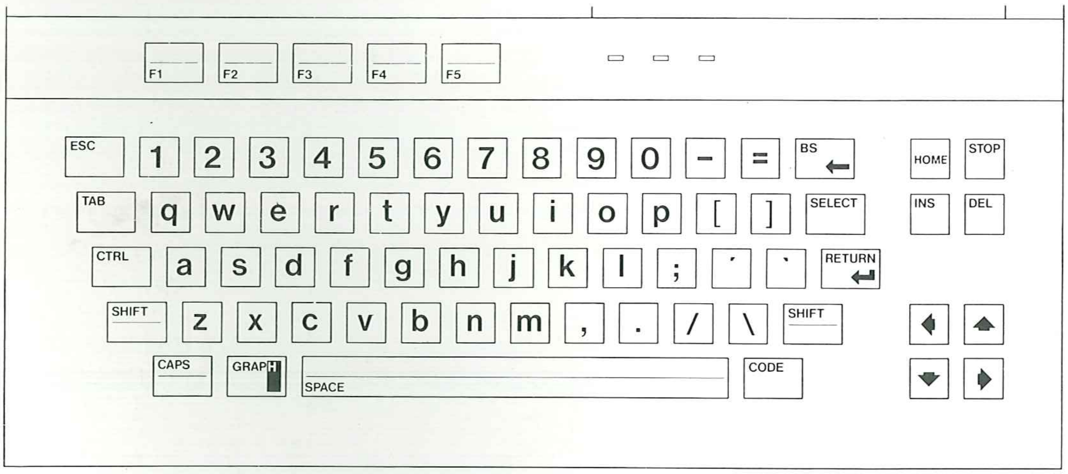
ABS	DATA	IF	NAME	SAVE
AND	DEF	IMP	NEW	SCREEN
AS	DEFINT	INKEY$	NEXT	SGN
ASC	DEFDBL	INP	NOT	SIN
ATN	DEFSNG	INPUT	OCT$	SOUND
AUTO	DEFSTR	INPUT$	OFF	SPACE$
BASE	DEFUSR	INSTR	ON	SPC
BEEP	DELETE	INT	OPEN	SPRITE
BIN$	DIM	INTERVAL	OR	SPRITE$
BLOAD	DRAW	KEY	OUT	SQR
BSAVE	DSKF	KILL	PAD	STEP
CALL	ELSE	LEFT$	PAIN	STICK
CDBL	END	LEN	PDL	STOP
CHR$	EOF	LET	PEEK	STR$
CINT	EQV	LINE	PLAY	STRIG
CIRCLE	ERASE	LIST	POINT	STRING$
CLEAR	ERL	LLIST	POKE	SWAP
CLOAD	ERR	LOAD	POS	TAB
CLOSE	ERROR	LOC	PRINT	TAN
CLS	EXP	LOCATE	PSET	THEN
COLOR	FIELD	LOF	PRESET	TIME
CONT	FILES	LOG	PUT	TROFF
COPY	FIX	LOPS	READ	TRON
COS	FN	LPRINT	REM	USING
CSAVE	FOR	LSET	RENUM	USR
CSNG	FRE	MAXFILES	RESTORE	VAL
CSRLIN	GET	MERGE	RESUME	VARPTR
CVD	GOSUB	MID$	RETURN	VDP
CVI	GOTO	MKD$	RIGHT$	VPEEK
CVS	HEX$	MKI$	RND	VPOKE
		MOD	RSET	WAIT
		MOTOR	RUN	WIDTH
		MKS$		XOR

Teclado

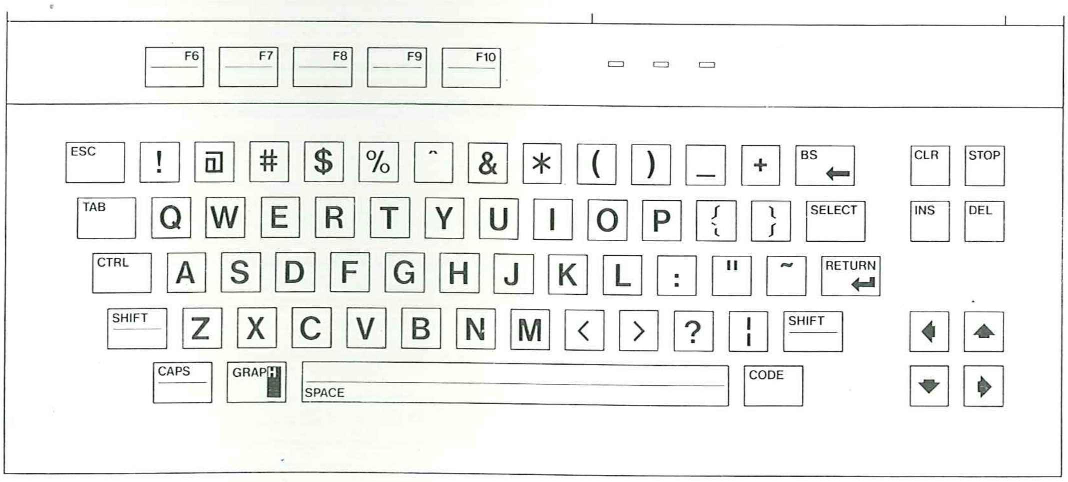
Apéndice G

Este apéndice es un resumen completo de todos los caracteres que pueden ser generados mediante el teclado, y enviados hacia el procesador interno.

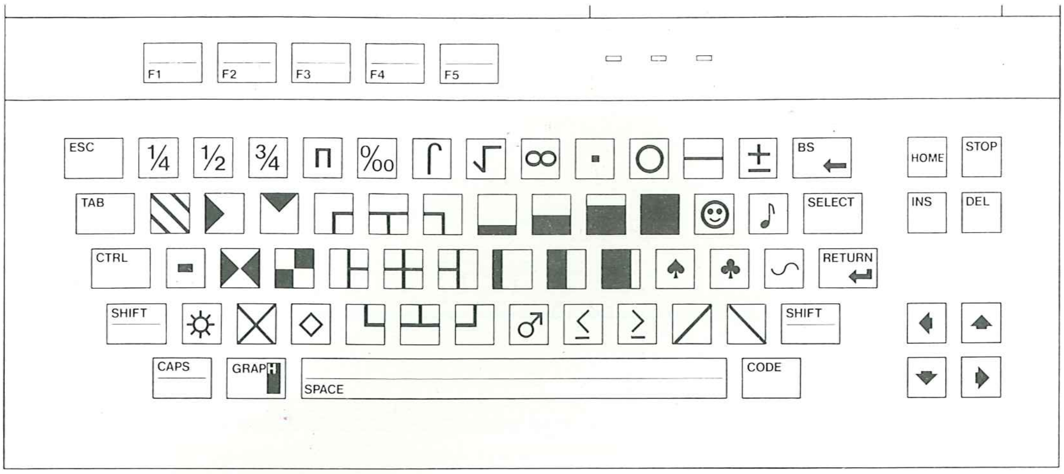
A. Caracteres prescritos.



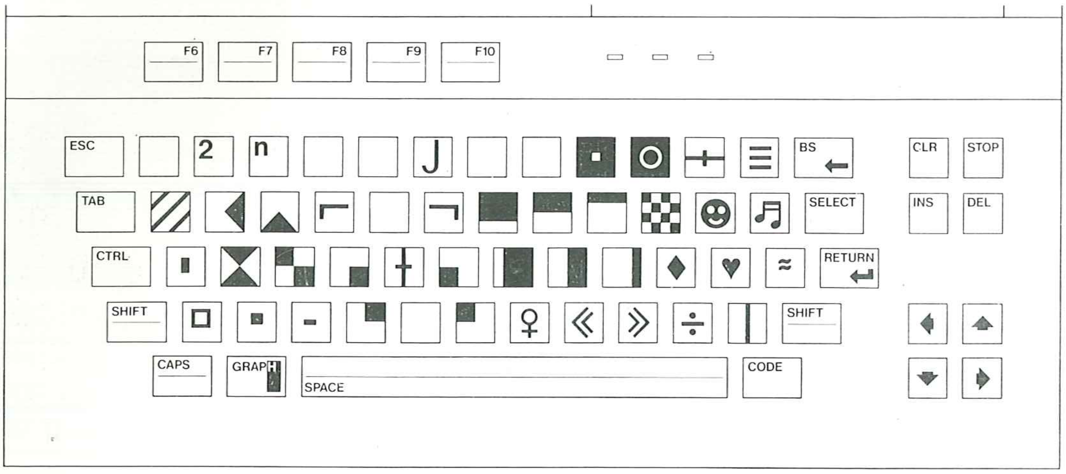
B. Caracteres alternativos, disponibles cuando se pulsa la tecla correspondiente, simultáneamente con la tecla de turno (**"SHIFT"**).



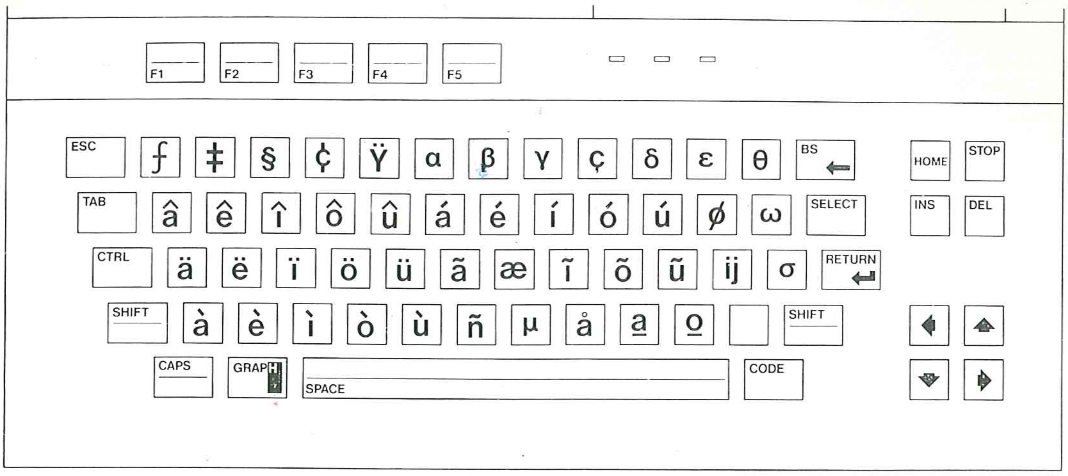
C. Caracteres prescritos gráficos, disponibles después de pulsar la tecla de gráficos ("**GRAPH**").



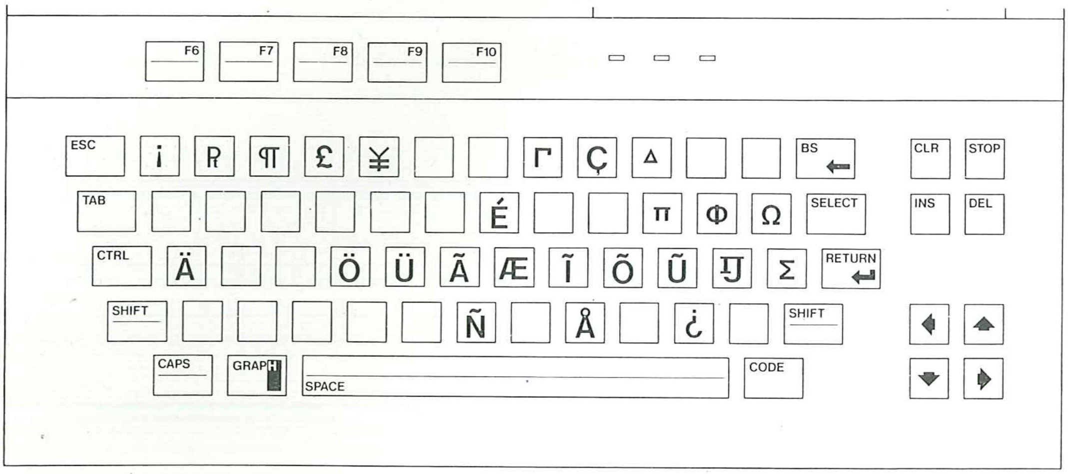
D. Caracteres alternativos gráficos, disponibles después de haber pulsado la tecla "GRAPH", y cuando se pulsa simultáneamente la tecla correspondiente y la tecla de turno ("**SHIFT**").



E. Caracteres prescritos especiales, disponibles después de haber pulsado la tecla de código (**"CODE"**).



F. Caracteres alternativos especiales, disponibles después de haber pulsado la tecla de código, y cuando se pulsa simultáneamente la tecla correspondiente con la tecla de turno (**"SHIFT"**).



Especificación técnica

Apéndice H

1. Circuitos integrados componentes:

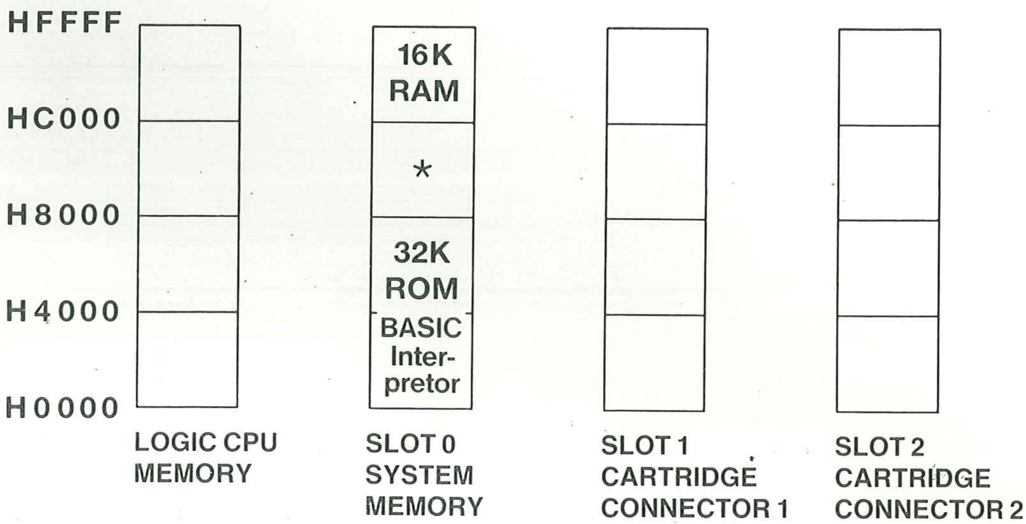
CPU Microprocesador Central
Z80A, 3.5 MHz.

VDP Procesador de Imágenes en Video
TI TMS-9929A o similar.

PSG Generador de Sonido Programable
GI AY-3-8910 o similar.

PPI Adaptador de Periféricos Programable
I 8255.

2. Composición de la memoria en **capas**.



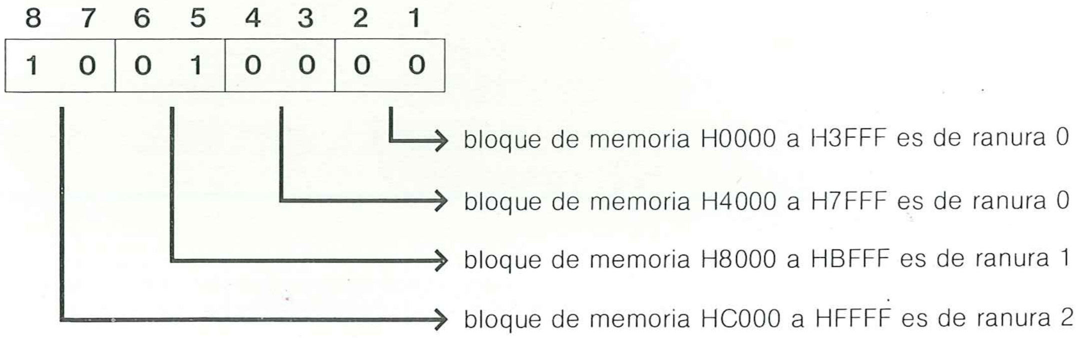
* Este área se completa con otras 16K RAM en los MSX 'home computer' de 32K RAM

El MSX 'home computer' también dispone de una memoria de escritura y lectura para video de 16K (véase Apéndice I de este manual).

Con memorias de 16K RAM, las direcciones de memoria entre &HC000 y &HC31F, y las direcciones entre &HF380 y &HFFF son usadas por la máquina como área de trabajo.

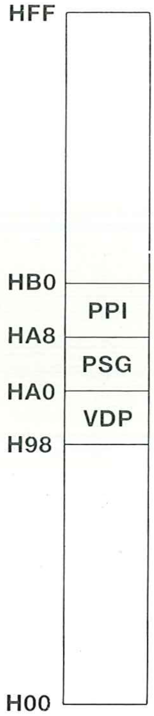
Con memoria RAM de 32K, las direcciones de memoria entre &H8000 y &H831F, y las direcciones de memoria &HF380 y &HFFFF son usadas por la máquina como áreas de trabajo.

El espacio **logical** de memoria, correspondiente a las direcciones mencionadas en las instrucciones del microprocesador, se 'concreta' en espacios físicos de memoria de las diversas ranuras **—slots—** provistas en la máquina. La manera en que los diferentes bloques de circuitos de memoria corresponden al espacio logical de direcciones, queda reflejado en el registro de selección de ranuras, que se encuentra en el portal A del PPI. Aquí hay un ejemplo:



La dirección del portal A puede encontrarse en la tabla de direcciones de entrada/salida.

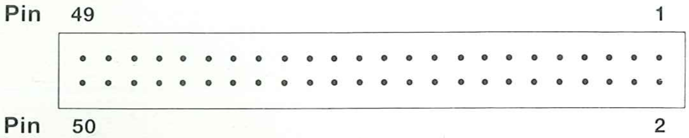
3. Tabla de direcciones de entrada/salida



ADRESS	R/W	DESCRIPTION	REMARK
HA8	W R	Port A data write Port A data read	8255 PPI
HA9	W R	Port B data write Port B data read	
HAA	W R	Port C data write Port C data read	
HAB	W	Mode set	
HA0	W	Adress latch	AY-3-8910
HA1	W	Data write	PSG
HA2	R	Data read	
H98	W R	V-RAM Data write V-RAM Data read	TMS 9929A VDP
H99	W R	Command, adress set Status read	

Con MSX-BASIC, es posible comunicarse directamente con los portales de entrada/salida de la máquina, usando la clave de instrucción "OUT" para sacar un octeto, la clave de instrucción "WAIT" para esperar a que haya un octeto dado en un portal de entrada, y la clave de función "INP" para examinar el octeto depositado en una entrada.

4. Conectores para cartucho.

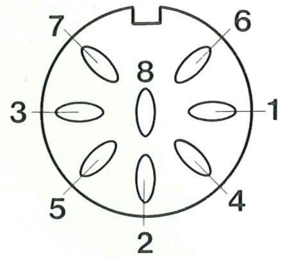


Pin	Name	I/O	Pin	Name	I/O
1	CSI	O	2	CS2	O
3	CS12	O	4	SLTSL	O
5	Reserve	–	6	RFSH	O
7	WAIT	I	8	INT	I
9	M1	O	10	BUSDIR	I
11	IORQ	O	12	MERQ	O
13	WR	O	14	RD	O
15	RESET	O	16	Reserve	–
17	A9	O	18	A15	O
19	A11	O	20	A10	O
21	A7	O	22	A6	O
23	A12	O	24	A8	O
25	A14	O	26	A13	O
27	A1	O	28	A0	O
29	A3	O	30	A2	O
31	A5	O	32	A4	O
33	D1	I/O	34	D0	I/O
35	D3	I/O	36	D2	I/O
37	D5	I/O	38	D4	I/O
39	D7	I/O	40	D6	I/O
41	GND	–	42	CLOCK	O
43	GND	–	44	SW1	–
45	– 5V	–	46	SW2	–
47	– 5V	–	48	+ 12V	–
49	SOUNDIN	I	50	– 12V	–

Pin	Name	Content
1	CS1	ROM addresses 4000 ~ 7FFF select signal
2	CS2	ROM addresses 8000 ~ BFFF select signal
3	CS12	ROM addresses 4000 ~ BFFF select signal (for 256k ROM)
4	SLTSL	Slot select signal
5	Reserve	Reserved signal line ~ use inhibited
6	RFSH	Refresh cycle signal
7	WAIT	CPU's WAIT request signal
8	INT	Interrupt request signal to CPU
9	M1	Signal expressing CPU fetch cycle
10	BUSDIR	This signal controls direction of external data bus buffer Cartridges are selected and L level is output from each cartridge at data transmission time
11	IORQ	I/O request signal
12	MERQ	Memory request signal
13	WR	Write timing signal
14	RD	Read timing signal
15	RESET	System reset signal
16	Reserve	Reserved signal line ~ use inhibited
17 ~ 32	A0 ~ A15	Address bus signals
33 ~ 40	D0 ~ D7	Data bus signals
41	GND	Signal ground
42	CLOCK	CPU clock 3.579545MHz
43	GND	Signal ground
44, 46	SW1, SW2	For insertion/removal protect
45,47	+ 5V	+ 5V power source
48	+ 12V	+ 12V power source
49	SOUNDIN	Sound input signal (– 5bdm)
50	– 12V	– 12V power source

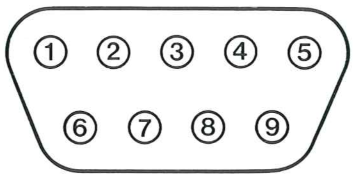
5. Conector de lecto-grabadora de datos.

Pin	Name	I/O
1	GND	
2	GND	
3	GND	
4	CMTOUT	O
5	CMTIN	I
6	REM +	O
7	REM –	O
8	GND	



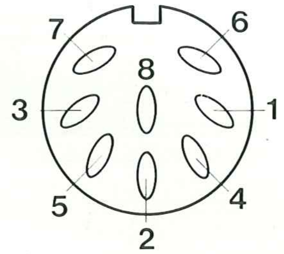
6. Conectores para joystick.

Pin	Name	I/O
1	FWD	I
2	BACK	I
3	LEFT	I
4	RIGHT	I
5	+ 5 V	
6	TRG 1	I/O
7	TRG 2	O
8	Output	O
9	GND	



7. Conector para monitor de video.

Pin	Signal PAL version	Signal RGB version*
1	+ 5 V	Status RGB
2	GND	GND
3	Sound	Blue
4	Luminance	Luminance
5	Comp. Video	Red
6	+ 12 V	+ 12 V
7	*)	Sound
8	*)	Green



*) RGB in french version only

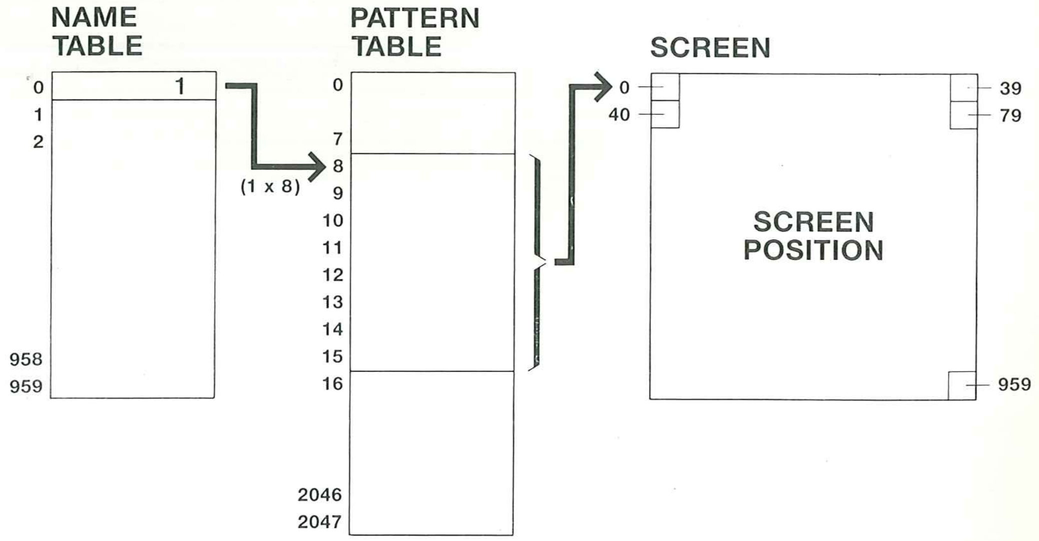
Procesador de imágenes de video (VDP)

Apéndice I

El MSX 'home computer' está equipado con un circuito integrado procesador de imágenes de video (VDP) que provee toda la información necesaria para producir en pantalla una imagen dada. Para conseguirlo, el VDP dispone de una memoria de lectura-escritura (RAM) de 16K. La información almacenada en dicha memoria de video, puede ser examinada mediante la clave de función "VPEEK", y puede almacenarse directamente cualquier valor mediante la clave de instrucción "VPOKE". Todas las instrucciones que permiten exponer datos, tal como la clave "PRINT", y todas las instrucciones de gráficos, tal como la clave "DRAW", son procesadas automáticamente por MSX-BASIC y alojadas en las celdillas de la memoria RAM de video. Todo lo que aparece en la pantalla es fiel reflejo de lo almacenado en la memoria RAM de video en forma de listas y tablas, que se tratan de diferente manera según el modo de gestión de la pantalla.

Modo 1 de texto

En este modo de gestión de pantalla, la imagen se compone a partir de una gradilla con 24 líneas horizontales y 40 líneas verticales, que determinan 960 mallas, o **posiciones.** Con este propósito se manejan dos tablas en la memoria RAM de video, llamadas: Tabla **nombres,** y tabla **tipos.** La tabla nombres contiene todos los códigos de los caracteres que han de aparecer en pantalla, ordenados según sus posiciones. En la tabla tipos se reservan 8 octetos por cada uno de los 'nombres' de la tabla nombres, para determinar cuál es la forma o aspecto con que ha de exponerse en pantalla dicho carácter. Es la posición en la tabla nombres la que indica la posición dentro de la gradilla de pantalla.



El lugar de comienzo de las tablas en la memoria RAM de video está alojado como valor de la variable sistemal "BASE(n)". Variando el argumento se obtienen las diversas direcciones: "BASE(0)" da la dirección de la tabla NOMBRES, mientras que "BASE(2)" corresponde a la tabla TIPOS.

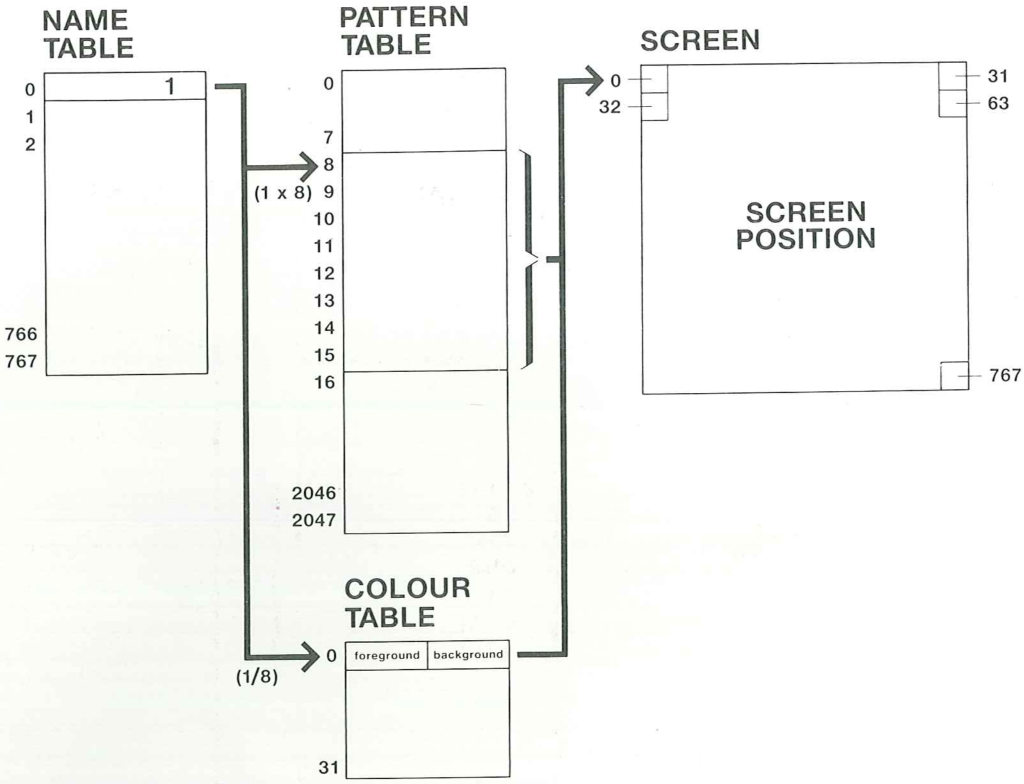
```
NEW
10 SCREEN 0:WIDTH 40
20 PRINT "HOLA"
30 PRINT BASE(0),BASE(2)
40 A=VPEEK(BASE(0)):PRINT A
50 B=A*8:FOR I=B TO B+7
60 C$="00000000"+BIN$(VPEEK(BASE(2)+I))
65 PRINT RIGHT$(C$,8)
70 NEXT I
RUN
```

En el ejemplo anterior, la primera dirección de las tablas NOMBRES y TIPOS se expone en la línea 30. El código del carácter (el 'nombre') letra H, se expone en la línea 40, como corresponde. Luego, se reproduce el perfil en binario (el 'tipo') de dicha letra H. Compara estos 8 trazos con la forma de la letra H que aparece en el Apéndice E.

Modo 2 de texto

En este modo de gestión de pantalla, la imagen se compone según una gradilla de 24 x 32 líneas, dando 768 mallas o posiciones. Con este propósito se manejan tres tablas en la memoria RAM de video: la tabla NOMBRES, la tabla TIPOS y la tabla COLORES. Las tablas NOMBRES y TIPOS funcionan de manera similar a la del modo 1 de texto. La tabla COLORES contiene 32 diferentes combinaciones de color del frente y de fondo.

Los caracteres 0 a 7 en la tabla NOMBRES usan la primera combinación de color en la tabla de COLORES. Los siguientes 8 caracteres usan la segunda combinación de color dentro de la tabla COLORES, y así sucesivamente. Toda combinación de color de la tabla COLORES se especifica mediante un octeto que ha sido subdividido en dos cuartetos. El primer cuarteto, los primeros 4 bits, indican el color del frente, el segundo cuarteto indica el color de fondo. La posición de la tabla NOMBRES en la memoria RAM de video, es el valor de la variable sistemal "BASE(5)", el comienzo de la tabla TIPOS es el valor de "BASE(7)" y la de la tabla COLORES es de "BASE (6)".



Cambiemos nuestro ejemplo anterior como sigue:

```
10 SCREEN 1:WIDTH 32
30 PRINT BASE(5),BASE(6),BASE(7)
40 A=VPEEK(BASE(5)):PRINT A
60 PRINT C$="00000000"+BIN$(VPEEK(BASE(7)+I))
80 D=VPEEK(BASE(6))
90 PRINT USING "##";D/16
100 PRINT USING "##";D MOD 16
RUN
```

La dirección inicial de las tablas NOMBRES, TIPOS y COLORES, se expone en este ejemplo, y luego se muestra el tipo o forma del carácter y el nombre o código de ese carácter. En este caso corresponde a la letra "H".

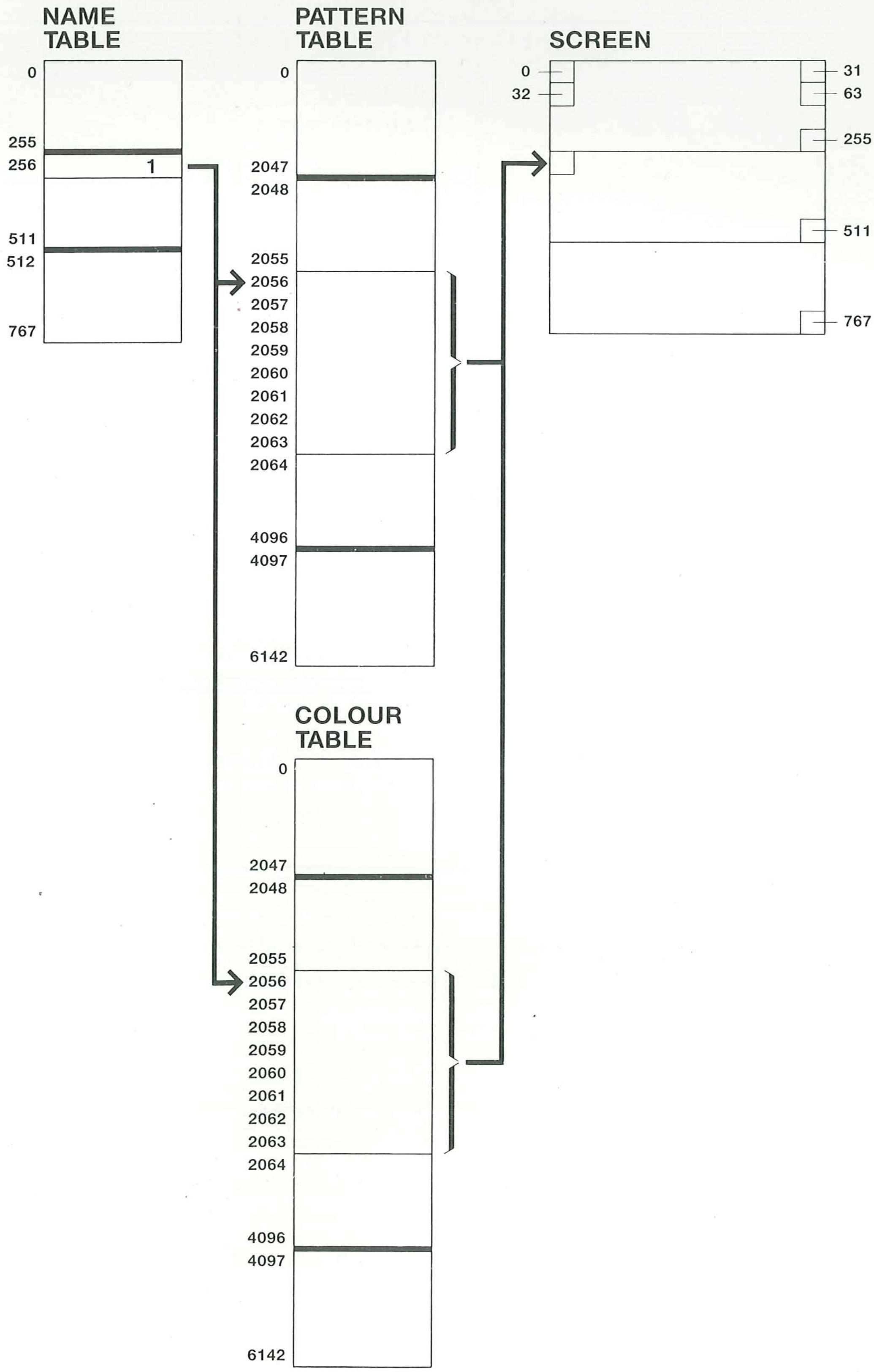
Cuando hayas examinado este ejemplo y lo hayas ejecutado, teclea:

```
SCREEN 0:WIDTH 40
```

Modo 1 de gráficos

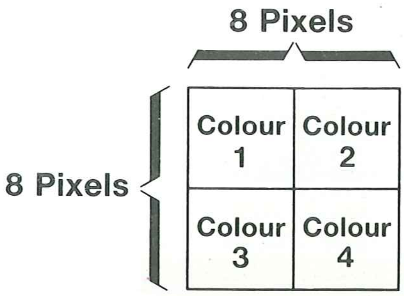
En este modo de gestión de pantalla la imagen está dividida en 3 x 8 líneas horizontales con 32 columnas por línea, lo que da un total de 768 **mallas** o posiciones. En este modo de pantalla también existe la tabla NOMBRES que señala correlativamente la posición ocupada en la pantalla. El número que figura en la tabla NOMBRES indica además dónde puede encontrarse el tipo o forma y el color del símbolo a exponer en pantalla. Por cada nombre de la tabla NOMBRES puede definirse un tipo con 8 x 8 motas dentro de la tabla TIPOS, con este propósito, la tabla TIPOS tiene una longitud de 6144 octetos (768 nombres x 8 octetos x nombre). Para cada uno de los octetos de la tabla TIPOS, puede definirse dentro de la tabla COLORES la correspondiente combinación de colores del frente y de fondo. Por lo tanto, la tabla COLORES tiene la misma longitud que la tabla TIPOS.

La posición de comienzo de la tabla NOMBRES en la memoria RAM de video es el valor de la variable sistemal de clave "BASE(10)" y la de comienzo de la tabla TIPOS DE "BASE(12)" y la de la tabla COLORES en "BASE(11)".



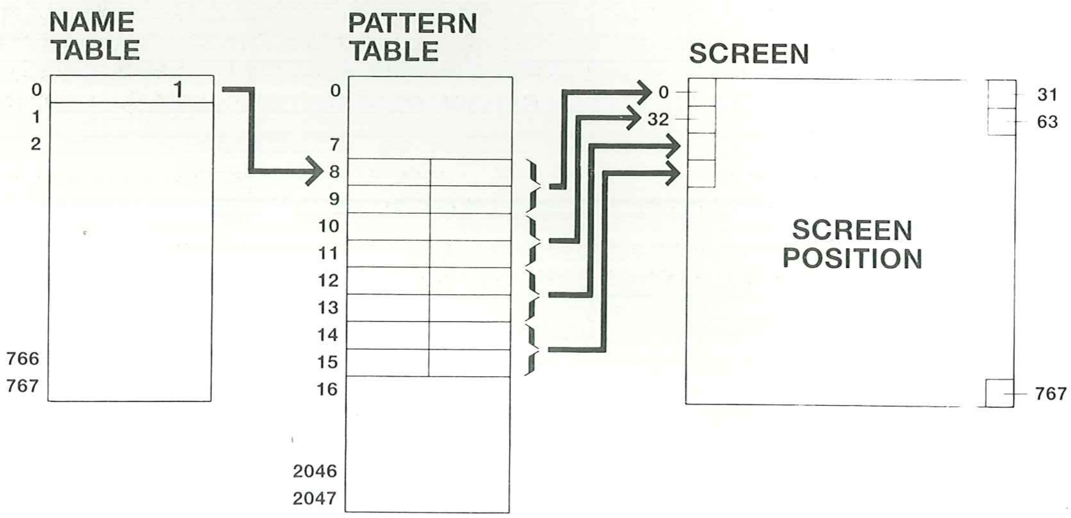
Modo 2 de gráficos

En este modo, la pantalla se compone según una gradilla de 24 líneas por 32 columnas, que da 768 posiciones o mallas; pero cada una consta de 4 cuadraditos con un tamaño de 4 x 4 puntos de imagen o motas. Cada uno de estos cuadritos contiene un color. En este modo de pantalla encontrarás una tabla NOMBRES que indica correlativamente la posición ocupada en pantalla. El número contenido en la tabla NOMBRES indica además dónde pueden encontrarse los 'colores' correspondientes a una posición de la pantalla, dentro de la tabla TIPOS.



Cada octeto de la tabla TIPOS ha sido dividido en dos cuartetos, y cada cuarteto indica mediante sus 4 bits un número de color. Cada nombre en la tabla NOMBRES requiere 8 octetos en la tabla TIPOS. Sólo 2 octetos del segmento de 8 octetos se usan para especificar la imagen en pantalla. Los primeros 2 octetos, en los grupos de segmentos de 8 octetos son usados para los nombres en la fila superior. La siguiente fila usa los octetos 3 y 4 en el segmento de 8 octetos. La siguiente fila usa los octetos 5 y 6, mientras que la última fila usa los octetos 7 y 8. Esta serie se repite para el resto de la pantalla.

La posición en la tabla NOMBRES en la memoria RAM de video, está contenida en la variable sistemal de clave "BASE(15)". La posición de la tabla TIPOS en "BASE(17)".



Sprites

Todos los 'sprites' están también contenidos dentro de la memoria RAM de video. La definición de la forma de un 'sprite' queda registrada dentro de la tabla TIPOS, mientras que la posición del 'sprite' en pantalla queda registrada en la tabla ATRIBUTOS, que determina el número y características de los 'sprites'.

La tabla TIPOS, que describe el 'sprite', consta de 8 octetos para 'sprites' de pequeño formato y 32 octetos para 'sprites' grandes. En la tabla ATRIBUTOS pueden definirse las posiciones y demás características de hasta 32 'sprites'.

Cada elemento de la tabla ATRIBUTOS consta de 4 octetos:

	7 6 5 4 3 2 1 0 Bit		
Byte 0	**Y-position**		
1	**X-position**		
2	**Sprite name**		
3			**Colour**

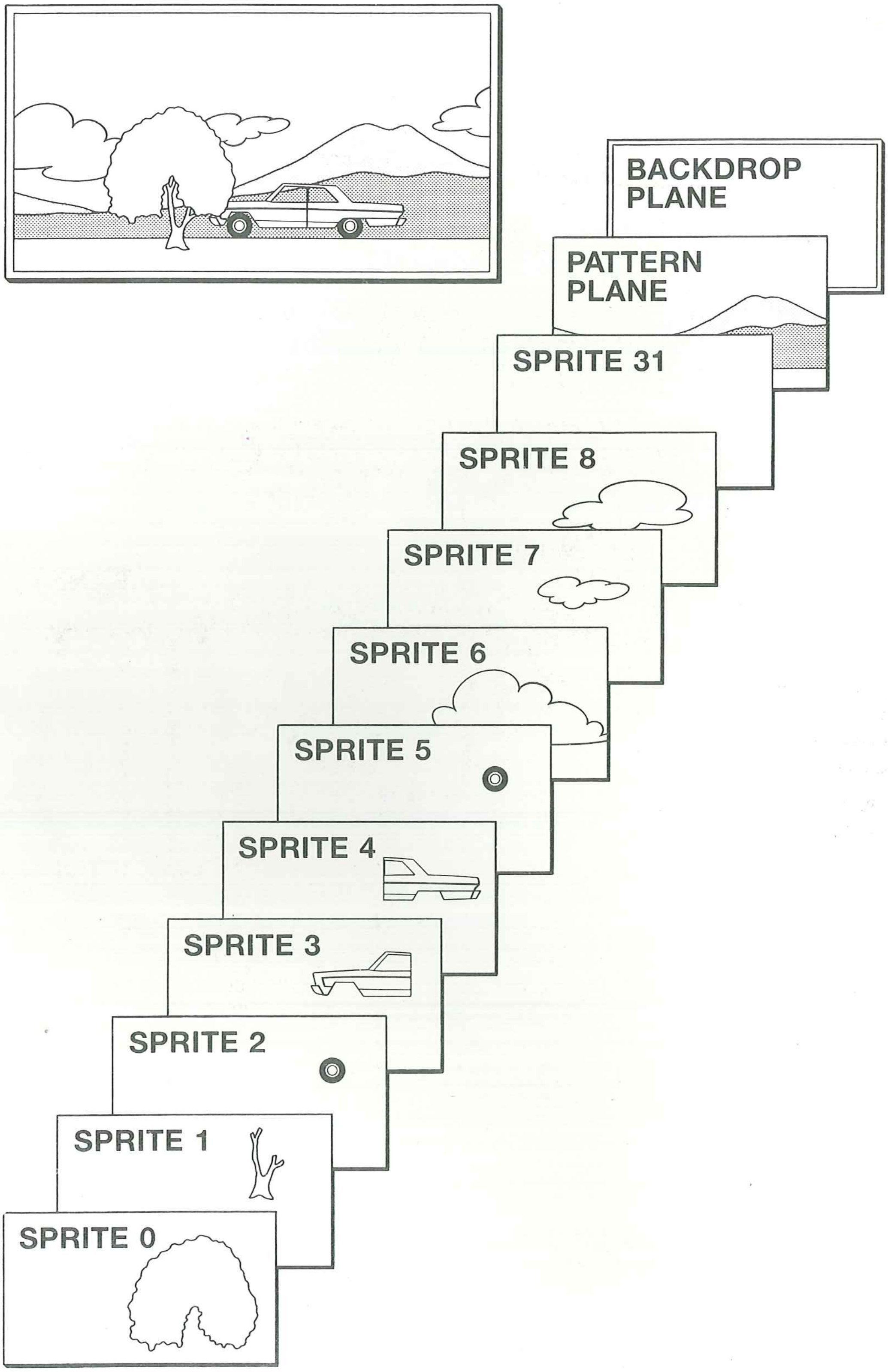
Puede colocarse un valor en el octeto que designe la posición Y, comprendido entre 0 y 255. Este valor determina la línea de la pantalla a la que será llevado el 'sprite', de acuerdo con lo siguiente:

0-191	designa el renglón o fila en pantalla
208	significa que el 'sprite' no aparece dentro de los límites de la pantalla.
224-255	representa las filas o renglones de pantalla de —31 a —1, respectivamente. Así es posible situar un 'sprite' por encima de la fila superior.

Puede asignarse un valor de 0 a 255 en el octeto que designa la posición X. Este valor determina la columna a que el 'sprite' será llevado. Si el bit 7 del octeto que designa el color es igual a 1, entonces la columna donde se llevará el 'sprite' iguala al valor de la posición X menos 32. Si la posición X es igual a 0 y sucede que el bit 7 del octeto que designa el color es 1, entonces el 'sprite' será llevado a la columna —32; con lo que se hace posible situar un 'sprite' a la izquierda de la primera columna de la pantalla.

La posición Y y la posición X mencionadas se aplican al punto de imagen, o mota situado en el ángulo superior izquierdo del 'sprite'.

El 'sprite' de prioridad máxima tiene como número de prioridad 0, y corresponde al primer elemento de la tabla ATRIBUTOS.


BACKDROP PLANE
PATTERN PLANE
SPRITE 31
SPRITE 8
SPRITE 7
SPRITE 6
SPRITE 5
SPRITE 4
SPRITE 3
SPRITE 2
SPRITE 1
SPRITE 0

En el dibujo anterior puedes ver un coche en un paisaje:

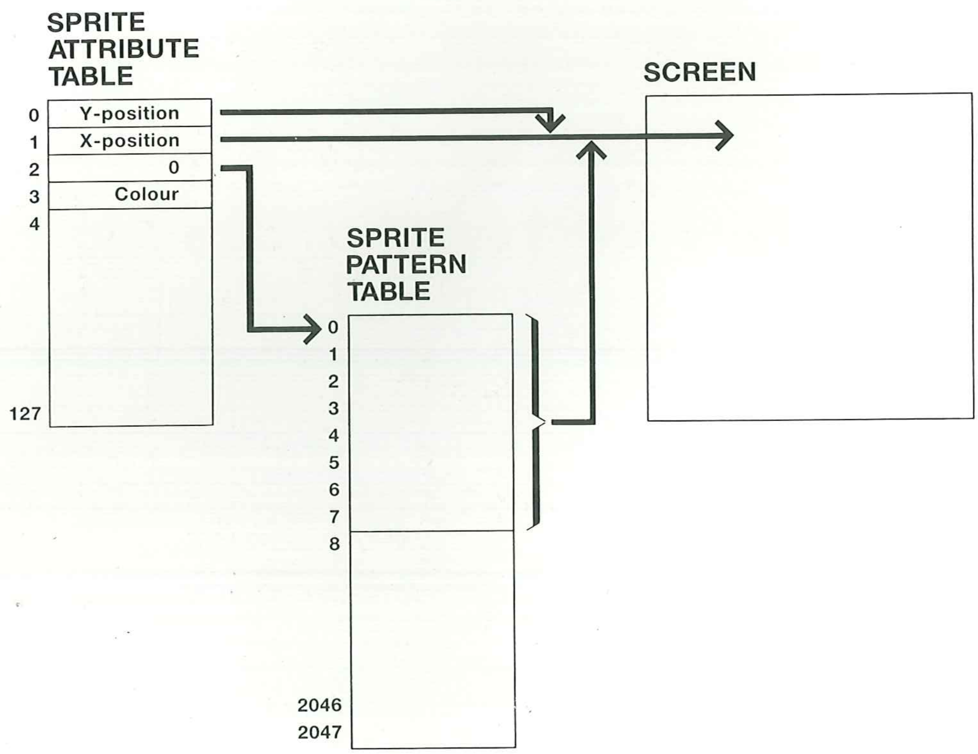
El árbol se compone con los 'sprites' 0 y 1.
El coche con los 'sprites' 2, 3, 4 y 5.
Las nubes con los 'sprites' 6, 7 y 8.

Si se asignan los números de prioridad, de acuerdo con los números de 'sprites' mencionados, los 'sprites' 0 y 1 (el árbol) tendrá prioridad sobre los 'sprites' 2 a 5 (el coche), con lo que el árbol aparecerá por delante del coche.

El octeto que designa el nombre de 'sprite' dentro de la tabla ATRIBUTOS es un número que indica donde encontrar el tipo, o forma del 'sprite' dentro de la tabla TIPOS.

La primera dirección de la tabla ATRIBUTOS de 'sprite', dentro de la memoria RAM de video, está contenida en la variable sistemal de clave "BASE(8)" en el modo 2 de texto, en la de clave "BASE(13)" en el modo 1 de gráficos, y en "BASE(18)" en el modo 2 de gráficos.

La primera dirección de la tabla TIPOS de 'sprite', está contenida en la variable sistemal de clave "BASE(9)" en el modo 2 de texto, en la de clave "BASE(14)" en el modo 1 de gráficos, y en la de "BASE(19)" en el modo 2 de gráficos.



Quita el programa que haya en memoria, tecleando el comando 'NEW' e intenta el siguiente ejemplo:

```
NEW
10 SCREEN 1:WIDTH 29
20 DATA 16,56,68,254,16,16,16,16
30 FOR I=0 TO 7
40 READ A:VPOKE(BASE(9)+I),A
50 NEXT I
60 PRINT:PRINT:PRINT
70 PRINT "UN SPRITE EN MODO 2 DE TEXTO"
```

```
80 VPOKE (BASE(8)+0),40
90 VPOKE (BASE(8)+2),0
100 VPOKE (BASE(8)+3),1
110 FOR I=0 TO 255
120 VPOKE (BASE(8)+1),I
130 NEXT I
140 GOTO 110
RUN
```

Este ejemplo muestra que es posible después de todo lo dicho, usar 'sprites' en el modo 2 de texto, si se comunica directamente con la memoria RAM de video. Pulsa simultáneamente CTRL y STOP para detener la ejecución del programa, y luego tecleas:

```
SCREEN 0:WIDTH 40
```

Registros El procesador de imágenes de video, dispone de 9 registros, numerados del 0 al 7 y el registro 8. Con MSX-BASIC el contenido de los registros está reflejado en el valor de las variables de clave "VDP(n)", siendo el argumento "n" un número entre 0 y 8. Exceptuando el registro 8, el valor de estas variables puede ser cambiado. El significado de los registros es el siguiente:

	0	1	2	3	4	5	6	7 Bit
Register 0	0	0	0	0	0	0	A	D
Register 1	1	0	E	B	C	R	S	M
Register 2	0	0	0	0	Name table			
Register 3	Colour table							
Register 4	0	0	0	0	0	Pattern table		
Register 5	0	Sprite attribute table						
Register 6	0	0	0	0	0	Sprite pattern table		
Register 7	Text colour 1				Text colour 2			
Register 8	T	N	V	5th Sprite				

ABC	
000 =	modo 2 de texto
100 =	modo 1 de gráficos
001 =	modo 2 de gráficos
010 =	modo 1 de texto

D = 0 significa que no hay entradas externas al VDP
D = 1 significa que sí hay entrada externa al VDP

E = 0 significa sin interrupciones del VDP
E = 1 significa con interrupciones del VDP

R ha sido reservada

S = 0 significa 'sprite' de 8 x 8 motas
S = 1 significa 'sprite' de 16 x 16 motas

M = 0 significa que los 'sprites' no están ampliados
M = 1 significa que los 'sprites' sí están ampliados

Tabla Nombres = primera dirección de la tabla nombres dentro de la memoria RAM de video.

Tabla Colores = primera dirección de la tabla colores dentro de la memoria RAM de video.

Tabla Tipos = primera dirección de la tabla tipos dentro de la memoria RAM de video.

Tabla Atributos de 'sprite' = primera dirección de la tabla atributos dentro de la memoria RAM de video.

Tabla Tipos de 'sprite' = primera dirección de la tabla que describe la forma de los 'sprites' dentro de la memoria RAM de video.

Color 1 texto = Color del frente (pluma) en el modo 1 de texto.

Color 2 texto = Color de fondo en modo 1 de texto.

T El banderín de estado "T" en el registro 8 está puesto a 1 al final del barrido de la última línea de la imagen activa. Es repuesto al valor 0 después de haber leído el estado del registro, o cuando el VDP es restaurado externamente.

N El banderín de estado "N" en el registro 8 es puesto a 1 cuando coinciden las posiciones de dos o más 'sprites'. La coincidencia ocurre si dos 'sprites' cualesquiera son llevados a la pantalla en una posición que se solapa. El banderín "N" se repone a 0 después de haber leído el registro 8 o cuando se restaura externamente el VDP. El registro 8 debiera leerse inmediatamente después de encender el ordenador, para asegurarse que el banderín de coincidencia está bajado. El VDP comprueba cada posición de mota ('pixel') para detectar la coincidencia durante la proyección de ese punto sin considerar dónde está situado en la pantalla. Eso ocurre cada 1/50 de segundo en el TMS9929A. Por tanto, cuando se mueven los 'sprites' más de una posición durante esos intervalos, es posible que los 'sprites' tengan múltiples motas que se solapan, o incluso que hayan pasado completamente uno por encima de otro sin que el VDP haya detectado la coincidencia.

V y 5.º 'sprite' El banderín de estado "V" en el registro 8 está puesto a 1 siempre que hay 5 o más 'sprites' dentro de la misma línea horizontal (líneas 0 a 192). Este banderín de estado se repone al valor 0 después de que se lea el registro 8, o cuando se restaura externamente el VDP. El número del quinto 'sprite' se coloca en los 5 bits inferiores del registro 8 cuando el banderín "V" está alzado y es válido siempre que el banderín "V" está a 1.

Las tablas en la memoria RAM de video pueden colocarse en una posición diferente a la asignada por MSX-BASIC, siempre que el usuario en su programa obedezca las siguientes reglas.

1. La tabla NOMBRES debe comenzar en la dirección 0 o en una dirección múltiplo de 1024.
2. La tabla COLORES debe comenzar en la direccón 0 o en direcciones múltiplo de 64.
3. La tabla TIPOS debe comenzar en la dirección 0 o en direcciones múltiplo de 2048.
4. La tabla ATRIBUTOS ('sprites') debe comenzar en la dirección 0 o en una dirección múltiplo de 128.
5. La tabla TIPOS de 'sprite' debe comenzar en la dirección 0 o en una dirección múltiplo de 2048.

El generador de sonido programable

Apéndice J

El MSX 'home computer' está equipado con un generador de sonido programable (PSG) que produce las notas emitidas. Con este propósito, el PSG dispone de 16 registros, a los que puede dotarse directamente de un valor mediante la clave de instrucción "SOUND". En contraste con la clave de instrucción "PLAY", el sonido continúa hasta que el registro en cuestión vuelve al valor 0 mediante otra instrucción de clave "SOUND".

Los registros tienen el siguiente significado:

	7	6	5	4	3	2	1	0 Bit
Register 0	Tone number channel A							
Register 1	0	0	0	0	Coarse tone A			
Register 2	Tone number B							
Register 3	0	0	0	0	Coarse tone B			
Register 4	Tone number channel C							
Register 5	0	0	0	0	Coarse tone C			
Register 6	0	0	0	Noise tone				
Register 7	0	0	RC	RB	RA	TC	TB	TA
Register 8	0	0	0	M	Volume A			

Register 9	0	0	0	M	**Volume B**
Register 10	0	0	0	M	**Volume C**
Register 11	**Form-length tone**				
Register 12	**Form-length coarse tone**				
Register 13	0	0	0	0	**Form number**

El tono de la nota emitida por los canales A, B y C se fija mediante el valor almacenado en los registros 0 a 5. El registro 7 determina el canal o canales por donde se emitirá el tono o el ruido generado por el circuito integrado mencionado. Si el registro 7 tiene como contenido el valor binario "00111110" significará que sólo se reproduce la nota a través del canal A.

El volumen de cada canal se determina mediante los valores contenidos en los registros 8 al 10. Cuando el bit M tiene el valor 1, significa volumen máximo.

Los registros 11 y 12 determinan la longitud de la envolvente (o perfil de la nota) elegida mediante el valor del registro 13. ,

El número de forma envolvente, contenido en el registro 13, tiene el mismo significado que el subcomando correspondiente de la instrucción de clave "PLAY".

El registro 14 se usa como portal A de entrada/salida, y el registro 15 como portal B de entrada/salida. Ninguno aparece en el diagrama.

Número de tono

El tono de cada nota viene fijado por los números contenidos en los registros 0 a 5, correspondiendo un par de registros para cada canal, y se determina de acuerdo con las fórmulas siguientes:

$$TP = \frac{CF}{16 \times TF} \qquad CT + \frac{T}{256} = \frac{TP}{256}$$

TP = Período de la nota
CF = Frecuencia de reloj (= 1.78977 mHz)
TF = Frecuencia de la nota
T = Tono fino (registros 0, 2, 4)
CT = Tono aproximado (registros 1, 3, 5)

Ejemplo: .

Una nota de 440 Hz, que es un tono de 440 ciclos, se determinaría:

$$TP = \frac{1.78977 \times 10\hat{\ }6}{16 \times 440} = 254 \qquad CT + \frac{T}{256} = \frac{254}{256}$$

CT = 0 T = 254

Si es para el canal A, sería el registro 0 = 254 y el registro 1 = 0.

Tono del ruido

Las características del ruido emitido se determina por su tono, de acuerdo con:

$$NP = \frac{CF}{16 \times NF} \qquad NT = NP$$

NP = Período del ruido
CF = Frecuencia del reloj (= 1.78977 mHz)
NF = Frecuencia del ruido
NT = Tono de la nota (registro 6)

Ejemplo: .

Un tono de ruido de 30 kHz, se determinaría como sigue:

$$NP = \frac{1.78977 \times 10^{\wedge}6}{16 \times (30 \times 10^{\wedge}3)} = 4 \qquad NT = 4$$

Registro 5 = 4

Longitud y forma de la envolvente

La longitud de la forma de envolvente viene determinada como sigue:

$$SP = \frac{CF}{256 \times SF} \qquad CT + \frac{T}{256} = \frac{SP}{256}$$

SP = Período de la forma
CF = Frecuencia de reloj (= 1.78977 mHz)
SF = Frecuencia de la forma
T = Valor de ajuste de la longitud de la forma (registro 11)
CT = Valor aproximado de la longitud de la forma (registro 12)

Ejemplo:

Una frecuencia de la forma (de envolvente de la nota) o 0.5 Hz se determina como sigue:

$$SP = \frac{1.78977 \times 10^{\wedge}6}{256 \times 0.5} = 14.000 \qquad CT + \frac{T}{256} = \frac{14.000}{256}$$

CT = 54 T = 176

Registro 11 = 176 y registro 12 = 54

Quita el programa que haya en memoria, usando el comando "NEW", y teclea el siguiente ejemplo:

```
10 FOR I=0 TO 13
20 SOUND I,0
30 NEXT I
40 SOUND 7,&H3E
50 SOUND 8,&H0F
60 FOR I=&H10 TO &HF0
70 SOUND 0,I
80 NEXT I
90 SOUND 6,&HOF
100 SOUND 7,&H07
110 SOUND 8,&H10
120 SOUND 9,&H10
130 SOUND 10,&H10
140 SOUND 12,&H40
150 SOUND 13,&H00
RUN
```

En las líneas 10 a 30, todos los registros del PSG son puestos al valor 0 mediante un bucle de clave "FOR...NEXT". En la línea 40 únicamente se prepara el canal A para emitir tonos.

En la línea 50 se determina el volumen de las notas emitidas por el canal A.

En el bucle de clave "FOR...NEXT", de las líneas 60, 70 y 80, se envían 3 notas sucesivas para el canal A, con tonos determinados.

En la línea 90 se determina el tono de ruido a usar.

En la línea 100 se determina el tono de ruido a usar.

En la línea 100 se indica que el ruido deberá reproducirse por los canales A, B y C.

En las líneas 110 a 130 se fija el máximo volumen para los tres canales.

En la línea 140 se determina la forma de envolvente a utilizar. En la línea 150 se determina la forma del ruido.

El sonido de las olas

El siguiente ejemplo da una imitación del sonido del mar:

```
NEW
10 FOR I=0 TO 13
20 SOUND I,0
30 NEXT I
40 SOUND 6,&H10
50 SOUND 7,&H07
60 SOUND 8,&H10
70 SOUND 9,&H10
80 SOUND 10,&H10
90 SOUND 12,&H40
100 SOUND 13,H0E
RUN
```

Todos los sonidos que oyes están producidos por el generador de sonido programable incorporado en tu 'home computer'. Después de haber ejecutado el programa, MSX-BASIC se coloca a nivel de comando mientras que el sonido continúa. Solamente parará cuando los registros 6 y 13 se coloquen a cero, o cuando pulses simultáneamente las teclas CTRL y STOP.

PHILIPS